

Spring 2010

A FPGA/DSP based ultrasound system for tumor detection

Ashish Ravindra Ratnakar
New Jersey Institute of Technology

Follow this and additional works at: <https://digitalcommons.njit.edu/theses>



Part of the [Electrical and Electronics Commons](#)

Recommended Citation

Ratnakar, Ashish Ravindra, "A FPGA/DSP based ultrasound system for tumor detection" (2010). *Theses*. 70.
<https://digitalcommons.njit.edu/theses/70>

This Thesis is brought to you for free and open access by the Theses and Dissertations at Digital Commons @ NJIT. It has been accepted for inclusion in Theses by an authorized administrator of Digital Commons @ NJIT. For more information, please contact digitalcommons@njit.edu.

Copyright Warning & Restrictions

The copyright law of the United States (Title 17, United States Code) governs the making of photocopies or other reproductions of copyrighted material.

Under certain conditions specified in the law, libraries and archives are authorized to furnish a photocopy or other reproduction. One of these specified conditions is that the photocopy or reproduction is not to be “used for any purpose other than private study, scholarship, or research.” If a user makes a request for, or later uses, a photocopy or reproduction for purposes in excess of “fair use” that user may be liable for copyright infringement,

This institution reserves the right to refuse to accept a copying order if, in its judgment, fulfillment of the order would involve violation of copyright law.

Please Note: The author retains the copyright while the New Jersey Institute of Technology reserves the right to distribute this thesis or dissertation

Printing note: If you do not wish to print this page, then select “Pages from: first page # to: last page #” on the print dialog screen

The Van Houten library has removed some of the personal information and all signatures from the approval page and biographical sketches of theses and dissertations in order to protect the identity of NJIT graduates and faculty.

ABSTRACT

A FPGA/DSP BASED ULTRASOUND SYSTEM FOR TUMOR DETECTION

**by
Ashish Ravindra Ratnakar**

This work presents a method of detection of size and location of tumor using ultrasound transmission. The system utilizes Quantitative Ultrasound (QUS) which means sending an ultrasound signal from a transmitter and receiving it at multiple receivers. This received signal is analyzed for echogenic as well as echolucent tumors to differentiate between the two along with non-tumorous sample and also for delay, signal distortion to determine the size/location of the tumor.

This analysis is further implemented using Field Programmable Gate Array (FPGA) and Digital Signal Processor (DSP) technologies. The proposed detection system utilizes Low Transient Pulse (LTP) technique. In this co-design architecture, the DSP carries out analysis of received demodulated signal at a lower speed while the FPGA runs at 62.5MHz for the generation of LTP signal and to demodulate bandpass ultrasonic signal sampled at 1MHz which interrupts DSP at every 1 μ S. This work elaborates the implementation of Quadrature Amplitude Modulation (QAM) receiver on FPGA for received signal from ultrasound detector. LTP is applied to the tumor samples through the transmitter and the received signal at ultrasonic receiver is passed through QAM to get different maxima (peaks) which are then further used for calculation of the location and subsequently, the size of the tumor using DSP.

This dual platform co-design demonstrates application of a FPGA/DSP platform for the generation of low transient pulse as well as processing of the received signal.

**A FPGA/DSP BASED ULTRASOUND SYSTEM
FOR TUMOR DETECTION**

**by
Ashish Ravindra Ratnakar**

**A Thesis
Submitted to the Faculty of
New Jersey Institute of Technology
in Partial Fulfillment of the Requirements for the Degree of
Master of Science in Electrical Engineering
Department of Electrical and Computer Engineering**

May 2010

Blank Page

APPROVAL PAGE

**A FPGA/DSP BASED ULTRASOUND SYSTEM
FOR TUMOR DETECTION**

Ashish Ravindra Ratnakar

Dr. Timothy Chang, Thesis Advisor
Professor of Electrical and Computer Engineering, NJIT

4/28/2010

Date

Dr. Mengchu Zhou, Thesis Co-advisor
Professor of Electrical and Computer Engineering, NJIT

4/28/2010

Date

Dr. Jie Hu, Committee Member
Assistant Professor of Electrical and Computer Engineering, NJIT

04/28/2010

Date

BIOGRAPHICAL SKETCH

Author: Ashish Ravindra Ratnakar

Degree: Master of Science

Date: May 2010

Undergraduate and Graduate Education:

- Master of Science in Electrical Engineering,
New Jersey Institute of Technology, Newark, NJ, 2010
- Bachelor of Engineering in Electronics,
University of Mumbai, Mumbai, India, 2004

Major: Electrical Engineering

*"To The Individual Tangible and Mental Sources of Inspirations from
Whom I Learnt to Live a Dream"*

*"To The Individuals Who Believe Serving for Human Beings is The Best
Prayer to God"*

ACKNOWLEDGEMENT

I am wholeheartedly thankful to my thesis advisor, Dr. Timothy Chang, without whom this research would not have been possible. His out-of-the-box thinking, enthusiasm, inspiration and in-depth knowledge were always helpful towards this achievement. I am grateful to him for showing faith in me even in case of discouraging moments.

I would also like to thank my thesis committee members, Dr. Mengchu Zhou and Dr. Jie Hu, for their immense encouragement and support. I am really thankful to them for being part of my thesis defense committee. I would especially like to thank Dr. Nirwan Ansari for coordinating everything with my committee members in absence of my advisor due to some unavoidable circumstances.

Dr. Biao Cheng (Bill) and Akash Mathur always supported me throughout this period the way they can. Their worthy contribution has influenced my research work.

Last but certainly not the least; I would like to thank my parents and my brother, Amey, who supported me in every possible way during my Masters degree.

TABLE OF CONTENTS

Chapter	Page
1 INTRODUCTION.....	1
1.1 Objective.....	1
1.2 Background.....	1
1.3 System Configuration.....	4
1.3.1 Physical Acoustics' DSP Starter Kit.....	5
1.3.2 Dalanco Spry's DSK Helper 1.....	6
1.3.3 Seiko D-Tran XY 3000.....	7
1.3.4 Physical Acoustics R15 α Ultrasonic Sensors.....	8
1.4 DSP and FPGA Considerations.....	8
1.4.1 C6416T as DSP.....	8
1.4.2 Xilinx Spartan IIE as FPGA.....	9
2 SYSTEM ANALYSIS.....	11
2.1 Ultrasound Systems.....	11
2.2 Systematic Approach.....	12
2.2.1 Ultrasonic Properties of Tumors.....	12
2.2.2 Preparation of Simulated Breast Sample.....	13
2.2.3 Assessment of Tumor Samples.....	13
3 SYSTEM IMPLEMENTATION.....	33
3.1 TMS320C6416T: DSK Implementation.....	33
3.1.1 TMS320C6416T Key Features.....	34
3.1.2 Memory Map of C6416 DSP.....	34

TABLE OF CONTENTS

(Continued)

Chapter	Page
3.1.3 Timers.....	35
3.1.4 Interrupts.....	36
3.1.5 Programming With Code Composer Studio.....	37
3.1.6 ADC Interrupt Processing.....	45
3.2 Xilinx Spartan-IIe: FPGA Implementation.....	47
3.2.1 Programming Steps.....	47
3.2.2 Simulation of Quadrature Amplitude Modulation.....	48
3.2.3 FPGA Programming.....	50
3.2.4 Quadrature Amplitude Demodulator.....	56
3.2.5 Entire FPGA System.....	68
4 EXPERIMENTS AND RESULTS.....	70
4.1 Execution Flow.....	70
4.2 Experimental Analysis.....	71
4.2.1 Calculation of Velocity of Ultrasound in Plain Sample.....	72
4.2.2 Echogenic Sample.....	72
4.2.3 Echolucent Sample.....	74
5 CONCLUSIONS AND FUTURE WORK.....	76
5.1 Conclusions.....	76
5.2 Areas of Improvement.....	77
5.3 Future Work.....	77

TABLE OF CONTENTS

(Continued)

Chapter	Page
APPENDIX A TIMER CONFIGURATION.....	78
APPENDIX B RESOLUTION.....	81
APPENDIX C SIMULATED BREAST SAMPLE PREPARATION.....	83
APPENDIX D EMIF CE SPACE REGISTER CONFIGURATION.....	84
REFERENCES.....	85

LIST OF TABLES

Table	Page
2.1 Calculations of Approximate Lengths for Echogenic Sample.....	24
2.2 Calculations of Approximate Lengths for Echolucent Sample.....	31
3.1 Variables associated with FPGA Configuration.....	40
3.2 Xilinx Spartan-IIe Family.....	47
3.3 Memory Interface between DSP and FPGA.....	54
3.4 Simulink Parameters for IIR Filter.....	62
4.1 Location of Echogenic Mass from Different Receiver Positions.....	72
4.2 Calculation of Different Parameters for Echogenic Sample.....	73
4.3 Distance of Echolucent Mass from Different Positions of Receiver.....	74
4.4 Calculation of Different Parameters for Echolucent Sample.....	75
D.1 CECTL0-3 Space Control Registers' Field Description.....	84

LIST OF FIGURES

Figure	Page
1.1 Experimental setup for tumor detection system.....	5
1.2 Block diagram of TMS320C6416T DSK.....	6
1.3 Seiko D-Tran X-Y 3000 Robot.....	7
1.4 Physical Acoustics" R15α ultrasonic sensor.....	8
1.5 Speed per dollar ratio for fixed-point package processors.....	9
2.1 Echolucent and echogenic masses.....	13
2.2 Echolucent and echogenic samples used for experiments.....	14
2.3 Low transient pulse drive signal generation.....	15
2.4 Quantitative Ultrasound (QUS) arrangement with Tx as transmitter and Rx as receiver.....	16
2.5 Low transient pulse drive signal input.....	17
2.6 Output for plain sample when receiver positioned at 180°.....	18
2.7 Output at receiver when positioned at 45° on a plain sample.....	19
2.8 Output at receiver when positioned at 90° on a plain sample.....	19
2.9 Output at receiver when positioned at 135° on a plain sample.....	20
2.10 Low transient pulse drive signal input.....	21
2.11 Output for echogenic phantom sample when receiver positioned at 25°...	21
2.12 Output for echogenic phantom sample when receiver positioned at 45°...	22
2.13 Output for echogenic phantom sample when receiver positioned at 90°...	22
2.14 Output for echogenic phantom sample when receiver positioned at 135°..	23
2.15 Output for echogenic phantom sample when receiver positioned at 180°..	23

LIST OF FIGURES

(continued)

Figure		Page
2.16	Computation of lengths for the echogenic sample using cosine law.....	24
2.17	Low transient pulse drive signal input.....	25
2.18	Output at receiver when the angle is 180°experimented on plain sample..	26
2.19	Output at receiver when the angle is 180°, experimented on echogenic Sample.....	26
2.20	Approximation of echogenic sample output using plain sample output....	27
2.21	Low transient pulse drive signal input.....	28
2.22	Output for echolucent phantom sample when the receiver is positioned at 20°.....	28
2.23	Output for echolucent phantom sample when the receiver is positioned at 45°.....	29
2.24	Output for echolucent phantom sample when the receiver is positioned at 90°.....	29
2.25	Output for echolucent phantom sample when the receiver is positioned at 135°.....	30
2.26	Output for echolucent phantom sample when the receiver is positioned at 180°.....	30
2.27	Computation of lengths for the echolucent sample using cosine law.....	31
3.1	TMS320C6416T architecture.....	33
3.2	Memory map of C6416T DSK.....	34
3.3	Interrupt service table for C6416.....	37
3.4	THS1206 configuration flow.....	42
3.5	Demodulated envelop at DAC2 output.....	45

LIST OF FIGURES

(continued)

Figure	Page
3.6 Demodulated envelope after simple moving average filtering at DAC2 output.....	46
3.7 Sample output and watch window output in Code Composer Studio.....	47
3.8 QAM Simulink design.....	49
3.9 Simulated outputs for input, receiver and demodulated signals.....	50
3.10 Register transfer level diagram for low transient pulse.....	51
3.11 Register transfer level diagram for Clock divider.....	52
3.12 Register transfer level for decoder to interface FPGA and DSP.....	55
3.13 Register transfer level for generation of control signal for DAC1.....	56
3.14 Register transfer level for generation of control signal for DAC2.....	56
3.15 Register transfer level for analog-digital converter.....	57
3.16 Matlab simulation for sine wave generator.....	58
3.17 Sine wave and cosine wave generator - ModelSim outputs.....	59
3.18 Register transfer level for sine and cosine generator.....	59
3.19 Direct-II form realization of IIR – transform domain filter.....	61
3.20 Magnitude response of IIR filter designed using parameters in Table 3.4.	62
3.21 Register transfer level for IIR Filter.	63
3.22 Simulation of the square root generator.....	64
3.23 RTL of high level module for square root generator.....	64
3.24 Register transfer level of Onemod module for square-root generation.....	65
3.25 Register transfer level of a Subtractor module.....	65

LIST OF FIGURES

(continued)

Figure		Page
3.26	Schematic of QAM demodulator.....	66
3.27	Output of receiver.....	67
3.28	FFT of output of receiver.....	67
3.29	Demodulated output of QAM.....	67
3.30	FFT of demodulated output of QAM.....	68
3.31	Interconnections of FPGA modules.....	69
4.1	Demodulated output at DAC2 with watch-window output in CCS when angle between transmitter and receiver was 180° - echogenic case.....	73
4.2	Demodulated output at DAC2 with watch-window output in CCS when angle between transmitter and receiver was 180° - echolucent case.....	74
A.1	Timer Period Register configuration.....	78
A.2	Timer Control Register configuration.....	79
A.3	Timer Count Register configuration.....	80
D.1	CECTL0-3 configuration registers.....	84

CHAPTER 1

INTRODUCTION

1.1 Objective

This thesis presents the design and the implementation of an ultrasound imaging technology based embedded system for the detection of a tumor and determination of its location and size.

It reviews different technologies for imaging for tumor detection. A Quantitative Ultrasound (QUS) based novel concept of Low Transient Pulse (LTP) is selected and implemented for better imaging and detection performance.

The FPGA/DSP based architecture is customized according the need of this proposed application, where FPGA stands for Field Programmable Gate Array and DSP for Digital Signal Processor. This architecture is originally developed for crack detection and thoroughly reviewed and checked whether it fits the need of the proposed system. After it is found to be reusable it is implemented to carry out different processes like the generation of LTP, demodulation of baseband signal and calculations of different parameters.

1.2 Background

This research mainly concentrates on breast tumor detection. Breast cancer is the second lead cause of cancer death for women in the USA and leading cause of cancer death for women worldwide. According to National Breast Cancer Coalition (NBCC), approximately 254,650 new cases of breast cancer may have been diagnosed in 2009 of

which 192,370 are invasive and 62,280 are of in situ breast cancer [1]. Hence, as per experts' suggestion, one must detect it early and cure it early.

There are many detection techniques based on Ultra-wideband (UWB) Microwave Imaging [2], Dynamic Contrast enhanced Magnetic Resonance Imaging (MRI) [3], Computer Tomography (CT). Most commonly used technique for early stage cancer detection is X-Ray Mammography but it has limitations of high false-negative rates (4%-34%) [4] and high false-positives (70%) [5], particularly in case of patients with radiographically dense tissues [6]. Also it is unsafe for repeated radiation especially for women under 40 years. MRI, which is more sensitive than X-Rays, also has higher rate of false-positives [7]. This thesis work applies low transient pulse [8] for the detection of tumor in simulated human breast.

Ultrasound Imaging is non-invasive/non-ionizing tool used in medical imaging. Quantitative Ultrasound System (QUS) has potential advantages over MRI, CT and X-Rays in terms of its cost, size, safety and detection resolution. In fact, the major concern of ultrasound imaging and QUS is their resolution. Usually, higher resolution is achieved by reduced penetration and higher cost but LTP compresses acoustic pulses for short duration to improve the resolution of imaging [8]. Another ultrasound guided microwave technique allows us to obtain tissue concentration information such as hemoglobin and water content and can recover permittivity and conductivity images of a target as small as 1.2mm [9] but it is quite expensive as it requires a tomographic imaging system. Some researchers suggest the usage of 3D ultrasound by comparing it with 2D ultrasound which is yet to be clinically tested [10]. *Sinha et al.* suggest the ways to reduce appreciable air gap near breast/paddle gap in B-mode ultrasound scanning. It

raises concerns over the necessity of reducing breast movement while traditional ultrasound scans [11].

Being a non-destructive testing (NDT) technology [12], ultrasound never permanently alters any characteristics of a target being tested but evaluates the different parameters. Being a non-invasive technology, it does not cause pain or discomfort. Ultrasonic waves emitted from a transducer into an object reflect back if there is an impurity or crack. This echo signal is analyzed for different parameters. LTP utilizes the same technique. It is a short duration pulse with negligible transient which in turn echoes out a high frequency signal when it comes across an impurity which can further be analyzed.

Unlike Rectangular Modulated Pulse (RMP), LTP does not require regenerative loops or modulating circuits to synthesize drive signal [8]. High speed logic gate arrays like FPGAs can generate short duration acoustic pulses like LTP, which in turn causes less phase interference and leads to better performance. A Digital Signal Processor (DSP) is a specialized microprocessor with an optimized architecture for digital signal processing. The purpose of FPGA/DSP combinational setup is to accelerate the execution speed of a task that could have been done on DSP alone thus unloading the DSP from the least important tasks [13]. There are many applications of FPGA/DSP environment like Predictive Current Control of Voltage-Source Inverters, Correlation-based LMS (CLMS) echo cancelling system, video processing, and radar signal processing [13-16]. In most of these applications, FPGAs are used for performing high frequency computations and DSPs for data processing. A task is partitioned between the two according to functionality and speed requirements [17-18]. This co-design could have been

implemented on FPGA alone as it is of low cost and high flexibility, but FPGA architectures have some significant disadvantages like increasing complexity and overhead [16].

To overcome these disadvantages of either technology, this system implements combination of the two to carry over their advantages with very little overhead. While FPGA performs high frequency tasks at 62.5MHz frequency, DSP carries out data processing at much lower frequency of just over 1MHz. A previously implemented system for crack detection using ultrasound [36] is studied for this purpose and modified to meet the requirements of the proposed tumor detection system.

1.3 System Configuration

The experimental setup of the proposed system is as shown in Figure 1.1, which consists of Spectrum Digital's DSP Starter Kit (DSK) based on TMS320C6416, Dalanco Spry's DSK Helper 1 daughter board consisting of Xilinx FPGA Spartan-IIIE (300K Gates), Physical Acoustics' Wide Bandwidth AE Amplifier, Seiko D-Tran XY 3000 X-Y Robot, a set of Physical Acoustics' ultrasonic transducer (R15 α) and Agilent Technology's 54622A oscilloscope. The two transducers are mounted using customized machinery on X-Y robot to carry out the tests for which output is viewed on oscilloscope.

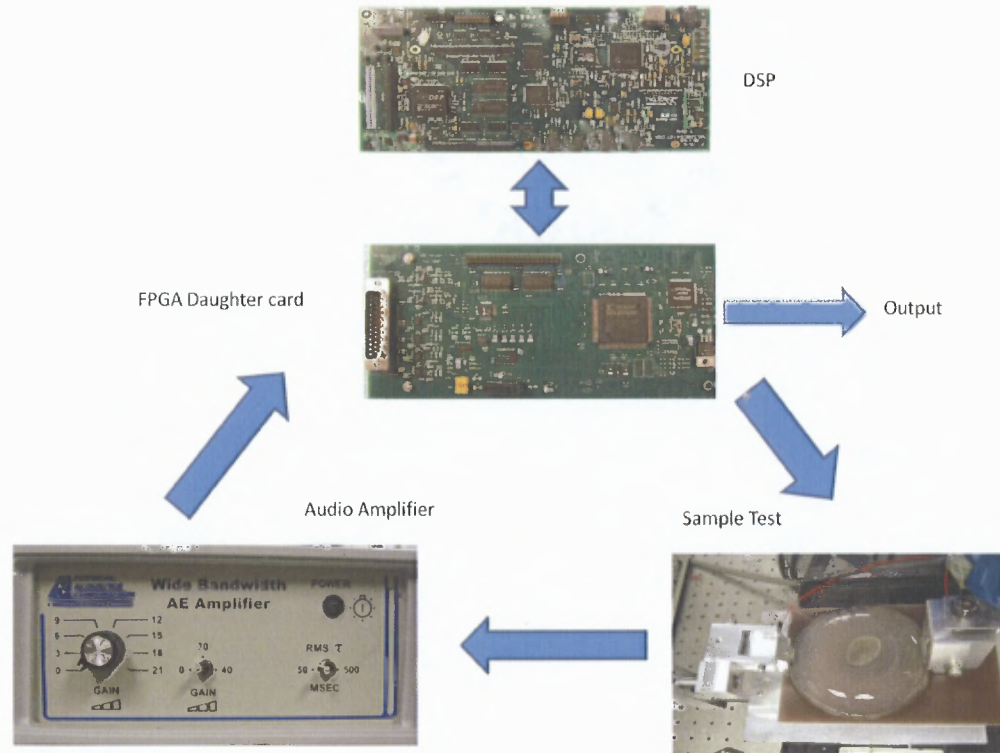


Figure 1.1 Experimental setup for tumor detection system.

1.3.1 Physical Acoustics' DSP Starter Kit

The Physical Acoustics' DSP starter kit is a low-cost standalone development platform that enables users to evaluate and develop applications on C6416 DSP [19]. This DSP operates at 1GHz clock with single +5V external supply connected to main input. The kit has onboard 512KB of flash ROM as well as 2M x 64 on board SDRAM. It provides three expansion connectors, i.e., Memory Interface, Peripheral Interface and Host Port Interface as shown in Figure 1.2, which can be used to accept plug-in daughter cards. Apart from this, there are many on-board components like AIC23 stereo codec that may not directly involved with our system. It also includes IDE called Code Composer Studio (CCS) that is used to program DSP using C programming language.

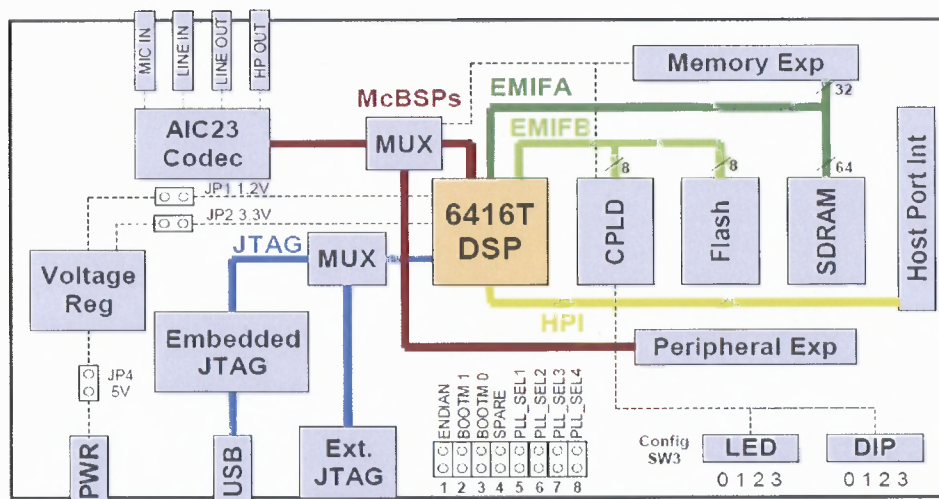


Figure 1.2 Block diagram of TMS320C6416T DSK.

Source: TMS320C6416T DSK Technical Reference, 508035-0001 Rev. A, November 2004

http://c6000.spectrumdigital.com/dsk6416/V3/docs/dsk6416_TechRef.pdf

1.3.2 Dalanco Spry's DSK Helper 1

This daughter board is responsible for LTP signal synthesis and the demodulation of receiver signal. The daughter board includes Xilinx Spartan-IIE 300K gates FPGA, 6 MSPS 4 channel 12 bit ADC – TI THS1206 and 25 MSPS+ 2 channel DAC – AD9765.

Spartan Iie is very low-cost second generation ASIC replacement technology from Xilinx with as high as 6,912 logic cells [20]. It requires internal power supply voltage (V_{CCINT}) of 1.8V for its operation. FPGA is customized by loading configuration data into internal static memory cell within configuration logic blocks (CLB). Hence, it can be reprogrammed over and over again. Stored values in these cells determine the logic functions and interconnections. This configuration data can be read from an external serial PROM in master serial mode or written into FPGA in slave serial, slave parallel or boundary scan mode [20].

TI THS1206 is CMOS, low power, 12-bit ADC with 68 dB SNR at 2MHz [21]. It runs at 1MHz clock from FPGA. AD9765 integrates two TxDAC cores, a voltage

reference and digital interface circuitry into a small 48-lead LQFP [22]. To meet our requirement, two channels of DAC run at two different rates: one with 62.5MHz for LTP drive signal and other at 1MHz that is baseband signal.

The configuration file for FPGA is generated using VHSIC Hardware Description Language (VHDL) [23]. This file determines the interactions of ADC, DAC and I/O with local bus. Xilinx provides IDE for developing of the configuration as well as schematic files.

1.3.3 Seiko D-Tran XY 3000

The working area for this robot is $300\text{mm} \times 300\text{mm} \times 100\text{mm}$. The angular movement is disabled. The speed is about 1500mm/sec. This robot can be programmed, but for our requirement, it was manually operated due to the irregular shape of samples. This robot is used to gain accuracy in our experimental results.



Figure 1.3 Seiko D-Tran X-Y 3000 Robot.

1.3.4 Physical Acoustics' R15 α Ultrasonic Sensors

R15 α is a general purpose, low-cost, 150 KHz resonant frequency acoustic emission sensor. Its peak sensitivity is 69dB with temperature range from -65 to 175°C. It has dynamic operating frequency range from 50 – 400 KHz [24].



Figure 1.4 Physical Acoustics' R15 α ultrasonic sensor.

1.4 DSP and FPGA considerations

1.4.1 C6416T as DSP

TMS320C6416T embeds high performance, 90nm, fixed-point C6416T DSP that can give performance up to 8000 million instructions per seconds (MIPS) at 1 GHz. It is also a very-long-instruction-word (VLIW) architecture developed by Texas Instruments (TI). It has low power consumption and hence, can be used to implement the proposed design [26].

This design needs maximum clock frequency of 62.5MHz for better resolution of LTP. C6416T can provide this clock for driver signal synthesis.

TMS320C64x embeds real time data exchange technology (RTDX) due to which one can view the application on host PC in real time. This technology works without interfering with the target application. On Windows host, it is handled by Component Object Modules (COM) through Code Composer Studio (CCS).

The speed/cost ratio in Figure 1.5 reveals that TMS320C64x platform is better than other competitive DSPs offered by Analog Devices, Freescale, Marvel and NEC [25].

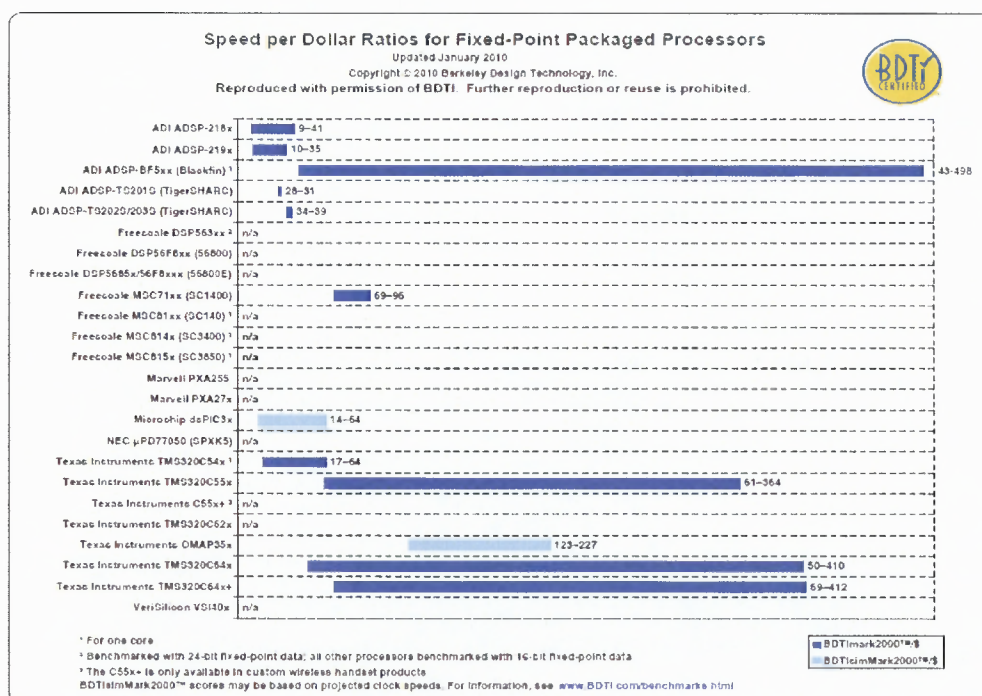


Figure 1.5 Speed per dollar ratio for Fixed Point Packaged Processors.

Source: BDTImark2000™/\$ and BDTIsimMark2000™/\$ - BDTI DSP Kernel Benchmarks, January 2010
 ©Berkley Design Technology, Inc.

1.4.2 Xilinx Spartan IIE XC2S300E as FPGA

Spartan-IIE FPGAs are typically used in high volume applications where the versatility of a fast programmable solution adds benefit. It shortens the product development cycle [20].

It offers on-chip synchronous single-port or dual-port RAM, DLL clock drivers, programmable set and reset on all flip-flops, fast carry logic and many extra features than its counterparts [20].

The daughterboard is designed in such way that it permits programming of FPGA through IDE of DSP, Code Composer Studio (CCS) which saves time from JTAG emulation.

CHAPTER 2

SYSTEM ANALYSIS

2.1 Ultrasound Systems

Ultrasound frequencies are cyclic sound pressure frequencies beyond human hearing abilities. The useful lower limit of ultrasound is 20 KHz. It is used in many applications typically to penetrate a medium and measure the reflection signature. This reflection signature reveals the details of inner structure of the medium and that is why it is used in Ultrasonography.

Ultrasonography is relatively inexpensive and portable than techniques like Magnetic Resonance Imaging (MRI) and Computed Tomography (CT) scanning. As mentioned earlier, sonography is a non-destructive technology (NDT) as it does not use any ionizing radiations causing chromosome breakage and cancer development. Typically, ultrasound energy produces a pressure wave through soft tissue and this pressure wave may cause microscopic bubbles in living tissue and distortion of cell membrane causing intracellular activity and ion fluxes [31]. However, ultrasound is considered safe when used at diagnostic power level by modern medical devices.

There are numerous biomedical applications of ultrasound including pregnancy detection, soft tissue imaging, detection of pelvic abnormalities, treatments of benign and malignant tumors using high intensity focused ultrasound (HIFU), cleaning teeth, treating cataract and so on. Other than biomedical applications, ultrasound is also used in welding, ultrasonic cleaning of metals, lenses and watches. It is also used for tracking and identification of objects.

2.2 Systematic Approach

Ultrasound-guided microwave imaging is now customary for breast tumor detection. It promises high contrast between tumor and normal soft tissues. A group of researchers suggests a new multimodality approach for high-resolution microwave imaging which uses a microwave imaging system and modified B-mode ultrasound [9]. This system can image a target of as small as 1.2mm using multimodality approach, which is impossible due to microwave scanning alone. Due to the usage of microwave scanning, the instrument becomes costly and tedious to work with.

The system described in this thesis improvises a simple approach for breast tumor detection using ultrasound. The analysis for the system is done in the following steps:

1. Ultrasonic properties of breast tumors are studied.
2. The simulated tumors of similar properties are used for sample preparation.
3. These samples are assessed and compared with each other as well as plain non-tumor sample using low transient pulse (LTP) drive to differentiate between them.
4. This analysis is further utilized to implement the system as described in Chapter 3.
5. The system implementation is tested for accurate results.

2.2.1 Ultrasonic Properties of Tumors

Tumors are classified into two categories [32].

- Benign – These are fluid-filled lesions, also known as cysts, which are echolucent (passing the sound but not reflecting it) in nature. They are non-carcinogenic and considered as the least aggressive.
- Malignant – These are solid lesions which are echogenic in nature. Malignant tumors are carcinogenic and considered as highly aggressive in nature. Some cancerous tumors are echogenic as well as echolucent in nature like Ovarian Cystadenocarcinoma, and Pancreatic Cystadenocarcinoma.

Human breast is as a whole echogenic structure and this echogenicity increases inside and near the tumor. In this thesis, it is assumed that breast tumors are hyperechoic (producing increasing amplitudes of waves in echo) in nature.

2.2.2 Preparation of Simulated Breast Sample

A customized echogenic as well as echolucent mass of size 2.5 x 3cm each (as shown in Figure 2.1) is ordered from Blue Phantom, WA, which will act as tumors. The simulated breast sample then can be prepared in seven days' time whenever required as explained in Appendix C. These samples, as shown in Figure 2.2, are usable for a week.



Figure 2.1 Echolucent and echogenic masses.

2.2.3 Assessment of Tumor Samples

Tumors are assessed for their properties using a pair of ultrasonic transceivers triggered by low transient pulse technique [8].

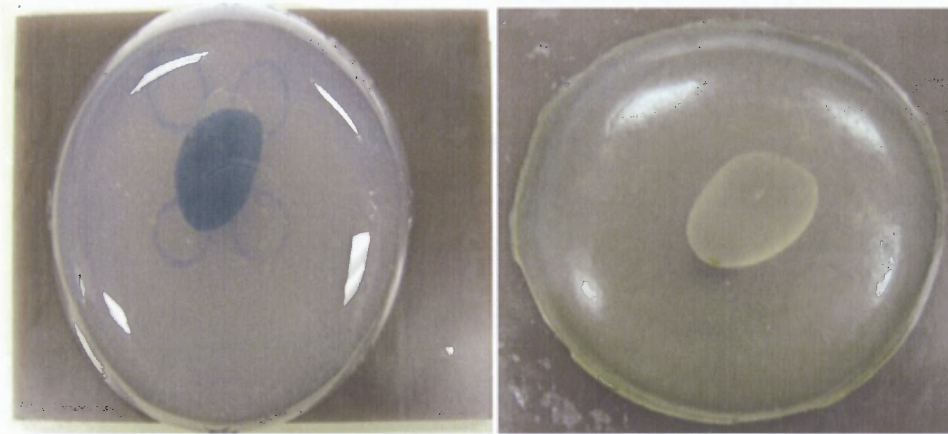


Figure 2.2 Echoluculent and echogenic samples used for experiments

Low Transient Pulse: Low transient pulse is a newly introduced technique that produces a short duration and low transient acoustic pulse by means of pre-shaping the transmitter excitation signal causing less phase interference and better performance.

For synthesis of drive signal, no modulation circuits or regenerative loops are necessary. Instead, it can be generated by using high speed logic gate arrays like FPGAs. The LTP method improves detection resolution by minimizing the aliasing of signals transmitted from soft and hard tissues.

The drive pulse of LTP is convolved with two impulses to generate necessary drive signal as shown in Figure 2.3. To produce continuous and integer oscillation cycles at the transmitter output, the pulse-width PW is equated to $t_2 - t_1$ to eliminate middle peak with amplitude $A_1 + A_2$.

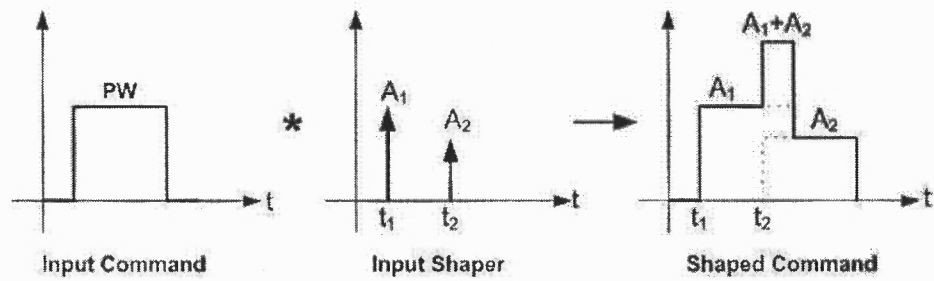


Figure 2.3 Low transient pulse drive signal generation.

A typical ultrasonic transmitter as well as receiver can be approximated by a second order underdamped transfer function [8] as below:

$$G(s) = \frac{\alpha s}{s^2 + 2\zeta\omega_n s + \omega_n^2}$$

The design parameters for two-impulse low transient pulse shaper can be summarized as below:

$$A_1 = \frac{1}{1 + M_p^n}, t_1 = 0$$

$$A_2 = \frac{M_p^n}{1 + M_p^n},$$

$$t_2 = \frac{n\pi}{\omega_n \sqrt{1 - \zeta^2}} = \frac{n\pi}{\omega_d},$$

$$M_p^n = \left(e^{\frac{\zeta\pi}{\sqrt{1 - \zeta^2}}} \right)^n, n = 1, 3, 5 \dots$$

ω_n = natural frequency of oscillations

ω_d = damping frequency of oscillations

ζ = damping factor

and A_1 and A_2 are impulse amplitudes that occur at t_1 and t_2 respectively as shown in Figure 2.3. The implementation of LTP is briefly explained later in Section 3.4.3.

Method of Analysis: It becomes necessary to standardize a method to analyze the sample and differentiate between the two. In this thesis work, the analysis is done in such a way that it imitates the B-mode Ultrasound medical imaging as shown in Figure 2.4.

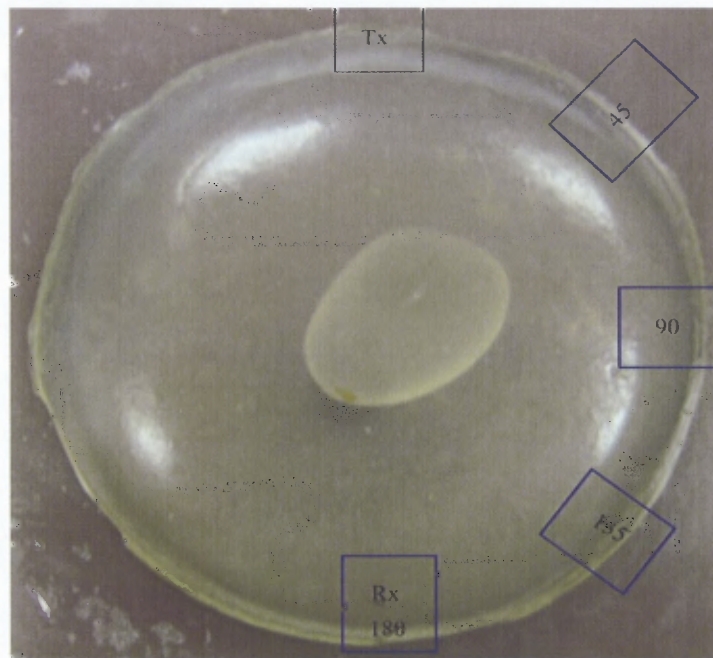


Figure 2.4 Quantitative Ultrasound (QUS) arrangement with Tx as transmitter and Rx as receiver.

In B-mode ultrasound, a linear array of transducers simultaneously scans a plane through the body that can be viewed as two-dimensional image on the screen [33]. During experimentation, the transmitter is kept at a fixed position and the receiver position is varied to fixed angles of 45° , 90° , 135° and 180° to observe the output. These observations are helpful to analyze and locate the tumors as well as to differentiate between them according to their properties.

Analysis of Plain Sample: The plain sample is analyzed to determine the velocity of ultrasound in gel. It is necessary for standardization while calculating and determining the location of a tumor while taking some delay into account in each echogenic as well as echolucent case. This is done by simply keeping two transducers 180° apart in a straight line and measuring the time taken by the signal to reach at the receiver after being transmitted and travelled through the gel medium. Having distance between the two transducers known and getting arrival time at the receiver, it was easy to calculate the velocity of ultrasound in gel by using the following equation.

$$v = \frac{d}{t} \quad (2.1)$$

where, d is the distance travelled through the gel medium and t is the time difference between transmitted LTP drive signal and first peak arrival at the receiver.

Figure 2.6 shows the output waveform at the receiver while LTP signal being transmitted is shown in Figure 2.5.

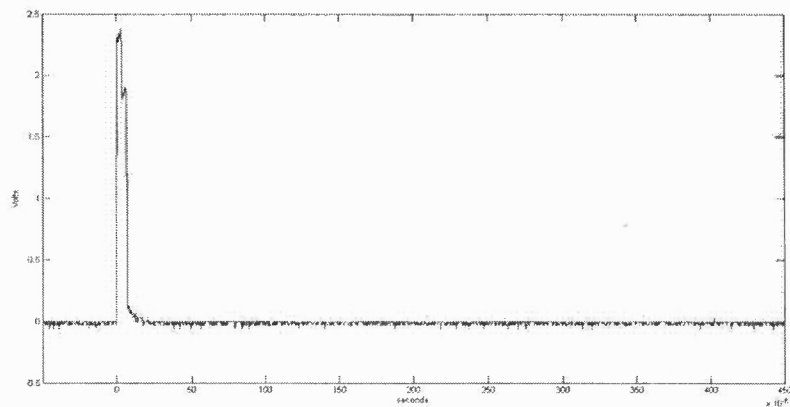


Figure 2.5 Low transient pulse drive signal input.

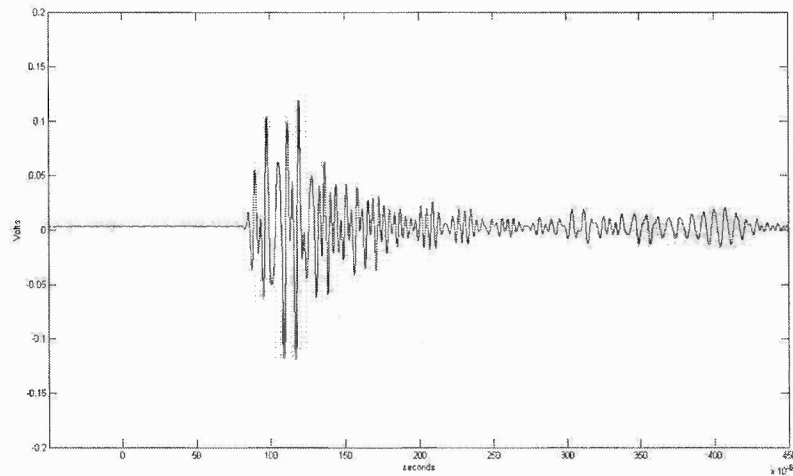


Figure 2.6 Output for plain sample when receiver positioned at 180° .

In a random sample, the distance between the transmitter and the receiver is 12.1cm and the time of first peak arrival is $78.25\mu\text{S}$. Hence, the velocity of ultrasound in the sample without tumor phantom is calculated as

$$U = 12.1\text{cm} / 78.25 \mu\text{S} = 1546.33 \text{ m/s}$$

This velocity of ultrasound can be used for calculations in echogenic and echolucent sample cases.

The response at different receiver positions is used to differentiate between the plain sample and tumor samples. Since a symmetrical sample is assumed, the output responses can be taken at 45° , 90° , 135° as shown in Figures 2.7 to 2.9.

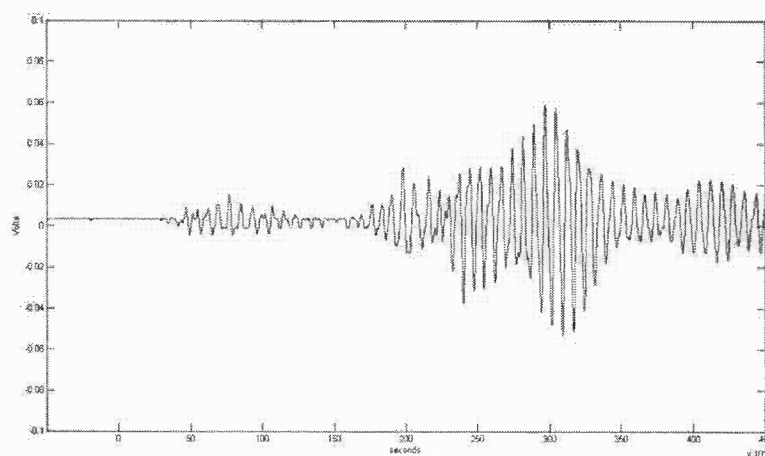


Figure 2.7 Output at receiver when positioned at 45° on a plain sample.

Figure 2.7 clearly shows transmitter-receiver coupling with multiple reflections. These multiple reflections make it hard to diagnose whether it is plain or tumorous.

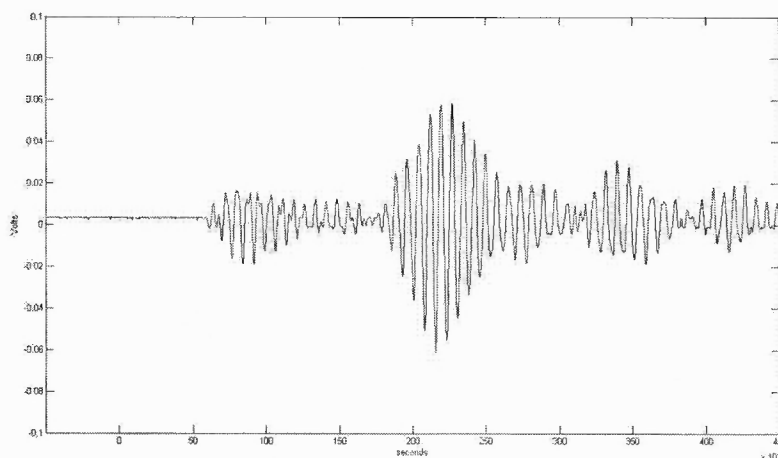


Figure 2.8 Output at receiver when positioned at 90° on a plain sample.

Figure 2.8 has the similar output as that of Figure 2.7. It shows reflection from the tumor and coupling between transmitter and receiver. There are at least 2-3 reflections which can be found from the surface.

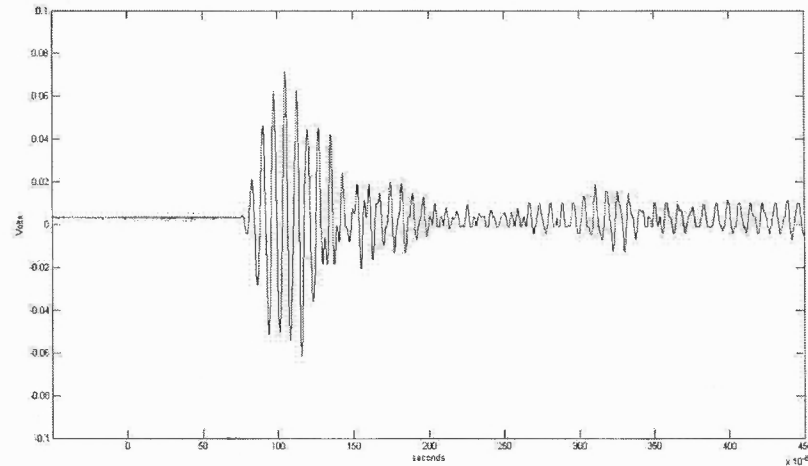


Figure 2.9 Output at receiver when positioned at 135° on a plain sample.

Figure 2.9 clearly indicates a similar waveform to that of 180° receiver location from the transmitter except that it has some reflections over the edges, which are mixed into the original transmitted wave.

Analysis of Echogenic Phantom Sample: As the output response characteristics of the samples are greatly dependent on user interaction, it becomes necessary to minimize user interference with the system. This is achieved by Seiko X-Y 3000 Robot and a set of customized assemblies. These assemblies permit us to keep the receiver and transmitter at the minimum of 20° apart from each other. This setup gives more accurate results which can be used for analysis further.

Similar to the plain sample case, the echogenic phantom sample is also tested for its characteristic response to low transient pulse input. The transmitter is kept at fixed position and the receiver is kept at different positions, i.e., 25° , 45° , 90° , 135° , and 180° . Similar to plain sample case, it is assumed to have bilateral symmetry. The response to the low transient pulse input is as shown in Figure 2.10.

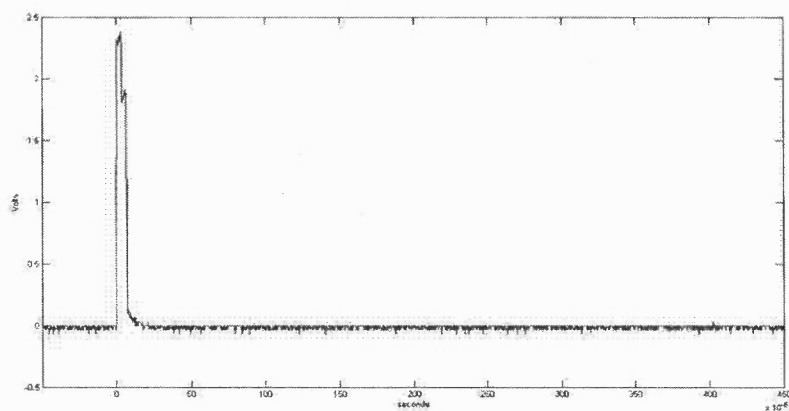


Figure 2.10 Low transient pulse drive signal input.

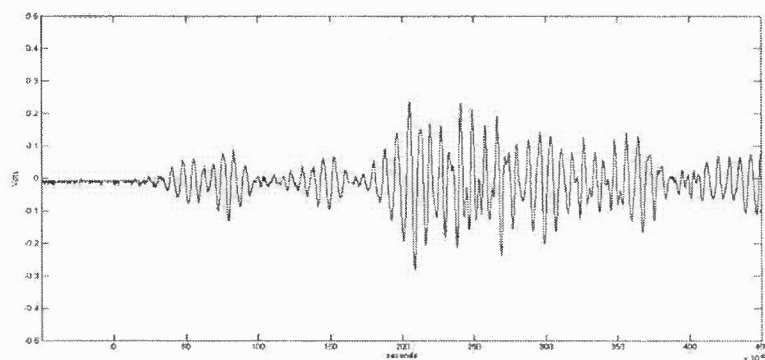


Figure 2.11 Output for echogenic phantom sample when receiver positioned at 25°.

Figure 2.11 clearly shows direct transmitter-receiver coupling due to their closeness as well as reflection from the tumor due to echogenicity. Other maxima are due to multiple reflections from the surface.

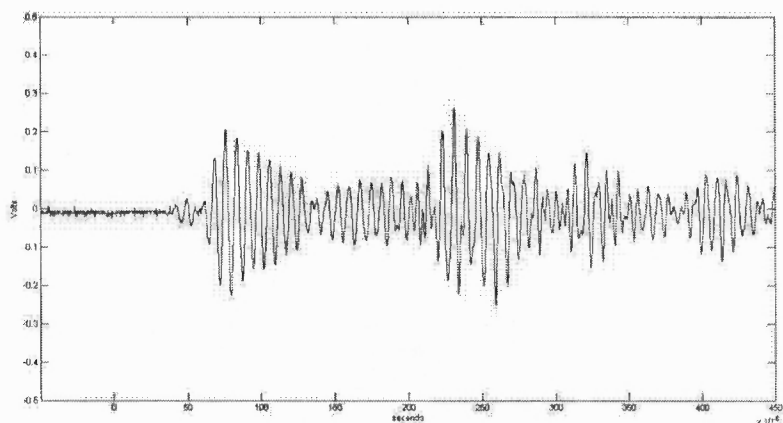


Figure 2.12 Output for echogenic phantom sample when receiver positioned at 45° .

Figure 2.12 reiterates the fact that echogenic tumor is located in close proximity of transmitter and receiver.

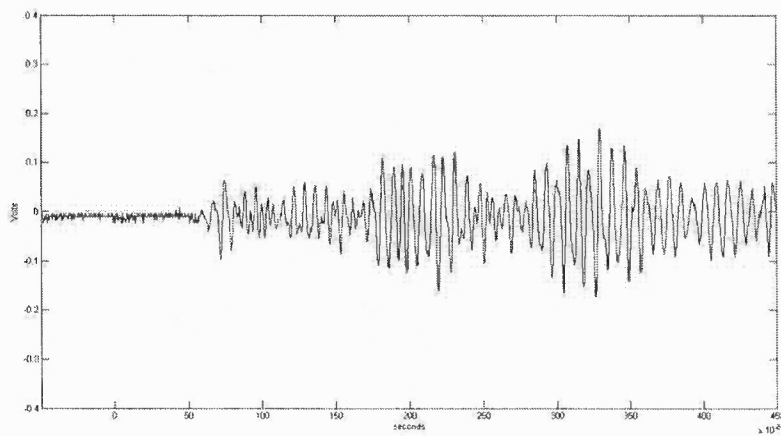


Figure 2.13 Output for echogenic phantom sample when receiver positioned at 90° .

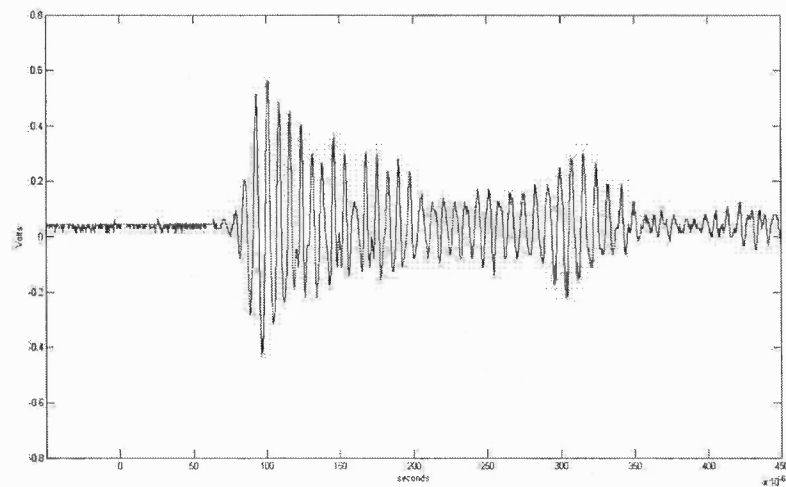


Figure 2.14 Output for echogenic phantom sample when receiver positioned at 135° .

Figure 2.14 indicates two close maxima which are barely differentiable and this is due to direct coupling of the transmitter to the receiver as well as reflection from echogenic phantom. The difference between the distances travelled by both the waves is very small.

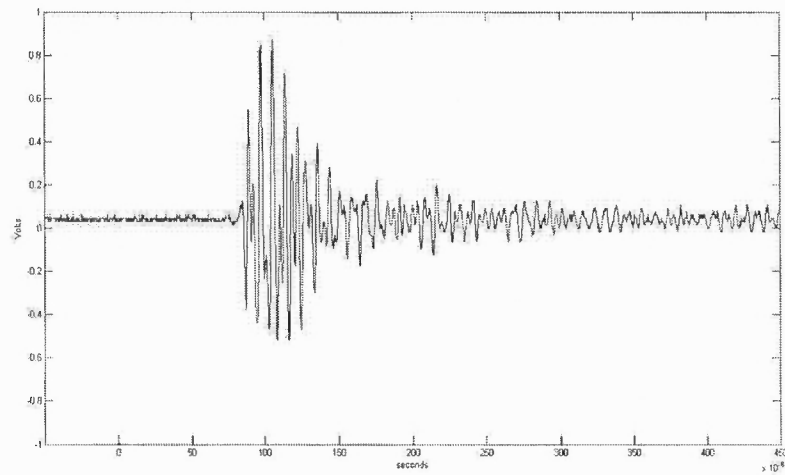


Figure 2.15 Output for echogenic phantom sample when receiver positioned at 180° .

Figure 2.15 shows clearly mixed up two or more waves hence, not differentiable.

Table 2.1 shows the arrival times for different peaks in the output where $\Delta T1$ is first peak arrival after the LTP signal is transmitted and $\Delta T2$ is the time difference between first peak and second distinct peak.

Table 2.1 Computation of Approximate Lengths for Echogenic Sample

Angle (in degrees)	$\Delta T1(\mu S)$	$\Delta T2(\mu S)$	$L1 = \Delta T1 * U$	$L2 = \Delta T2 * U$
25	32.375	28.625	5cm	4.43cm
45	52.375	77.5	8.1cm	11.98cm
90	58.25	48.25	9.01cm	7.46cm
135	69.375	75	10.73cm	11.6cm
180	75	68.875	11.6cm	10.65cm

The actual distance calculations are done by applying cosine law.

$$a^2 = b^2 + c^2 - 2bc \cos A \quad (2.2)$$

According to (2.2), the unknown distances between the transmitter and receiver locations are calculated as shown in Figure 2.16.

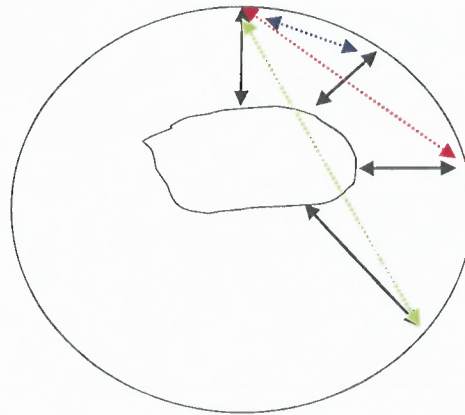


Figure 2.16 Computation of lengths for the echogenic sample using cosine law.

- The distance between Tx and Rx at 45° in blue is computed as 2.48cm.
- The distance between Tx and Rx at 90° in red is computed as 7.12cm.
- The distance between Tx and Rx at 135° in green is computed as 7.69cm.

It is observed that 45° case gives inconsistent results due to multiple reflections and hence, this case is not considered for further calculations. The following important observations are noted:

- The ratios of calculated values in Table 2.1 to the computed values using cosine law for 90° and 135° is nearly similar which is around 1.3.
- Although 0° case is impossible due to the assembly structure, the observations registered for the first peak values in case of 25° are nearly equal to the reflected path from the tumor. This case may detect presence of an echogenic tumor as the reflected signal is stronger and does not indicate transmitter-receiver coupling.

It can be proved that output received at the receiver in case of echogenic phantom sample is delayed and scaled version of the output received at the receiver in case of plain sample. Figures 2.17 to 2.20 show a set of readings which validates the same conclusion.

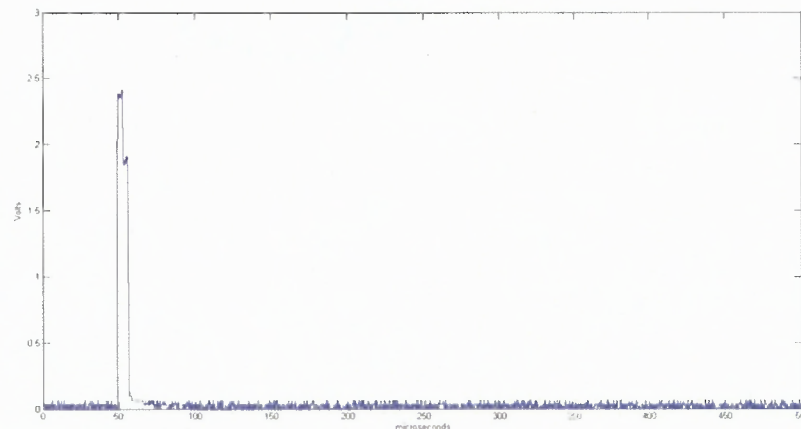


Figure 2.17 Low transient pulse drive signal input.

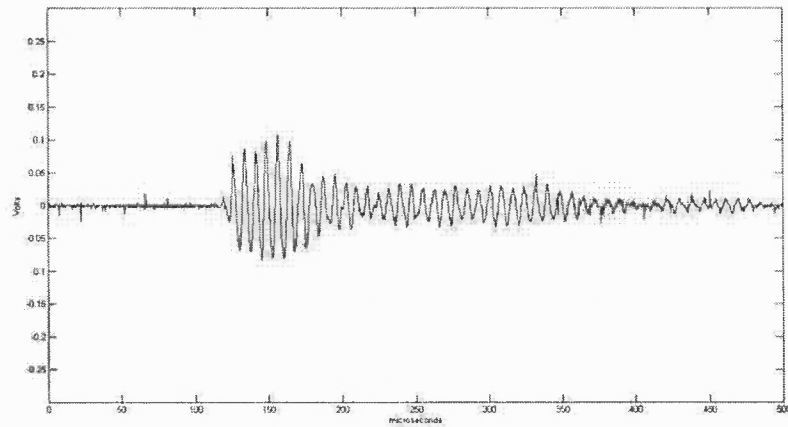


Figure 2.18 Output at receiver when the angle is 180° experimented on Plain Sample.

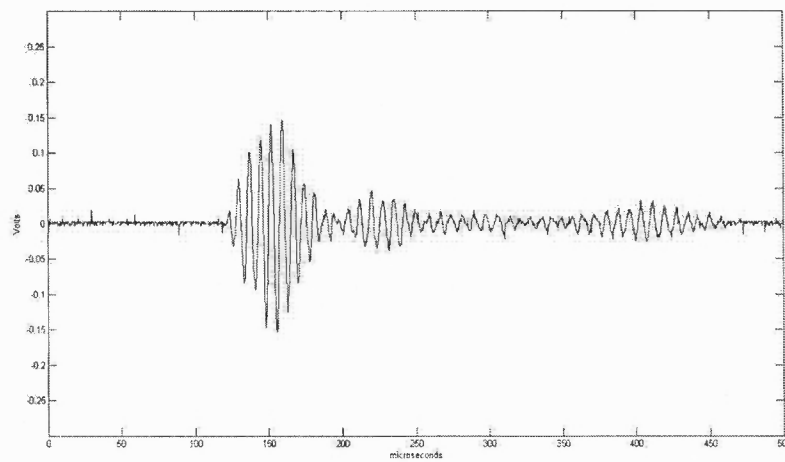


Figure 2.19 Output at receiver when the angle is 180° , experimented on echogenic Sample.

It is found using calculations that

$$d(Tx, Rx) = 10.2\text{cm.}$$

Time taken to reach Rx from Tx in plain sample = $t_1 = 67.8\mu\text{S}$.

Time taken to reach Rx from Tx in echogenic sample = $t_2 = 71.2\mu\text{S}$.

$$V_3(t) = A_1 V_1(t) + A_2 V_2(t - \tau) \quad (2.3)$$

where,

$V_3(t)$ = Velocity of Ultrasound in the echogenic sample.

$V_1(t)$ = Velocity of Ultrasound in the plain sample.

$V_2(t)$ = Velocity of Ultrasound in plain sample with alternate path

τ = Delay time.

The first part of (2.2) indicates plain sample signal and second part indicates delayed signal where $\tau = 3.4 \mu\text{s}$ and A_1 and A_2 are the scaling factors that are assumed to be 1. We obtain the following approximation as shown in Figure 2.20 which indicates that the assumption is correct.

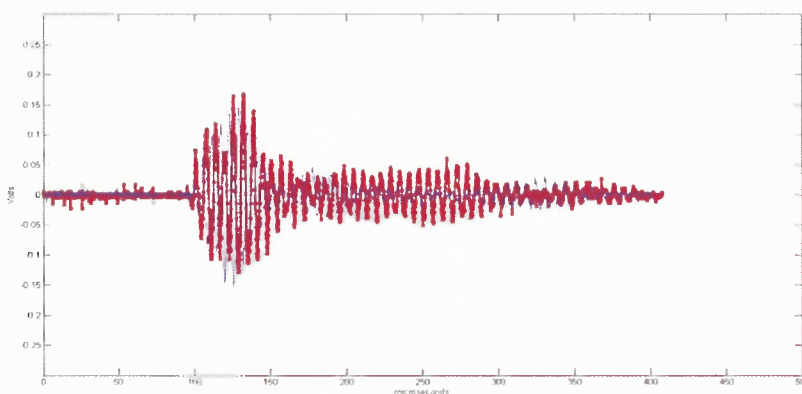


Figure 2.20 Approximation of echogenic sample output using plain sample output.

Analysis of Echolucent Phantom Sample: Similar to the echogenic phantom sample, the echolucent phantom sample is also tested at similar positions of the receiver with respect to the transmitter. The results are as explained in Figure 2.21 to 2.24. All figures are of the same scale.

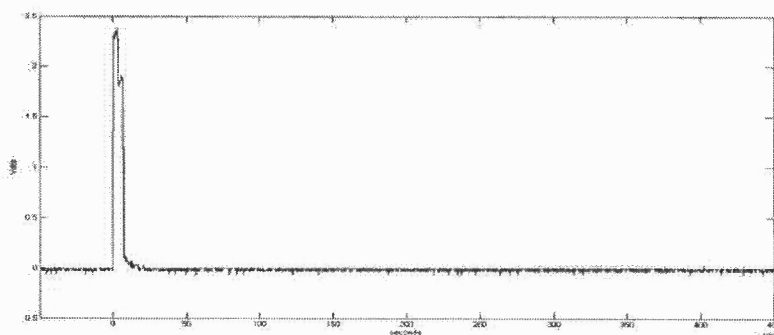


Figure 2.21 Low transient pulse drive signal input.

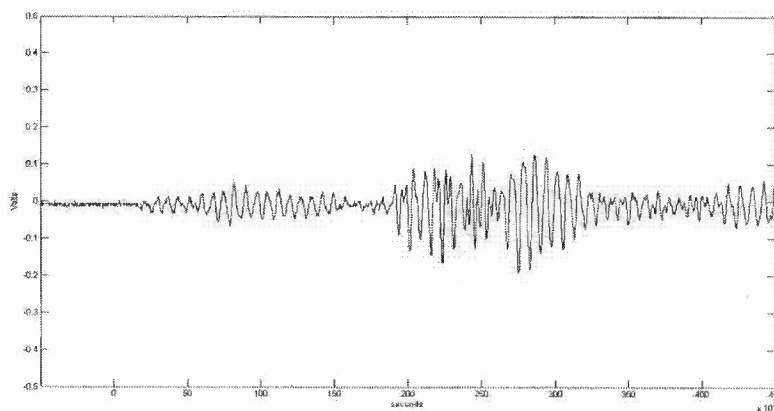


Figure 2.22 Output for echolucent phantom sample when the receiver is positioned at 20° .

In the 20° case, transmitter-receiver direct coupling is clearly visible immediately after input and reflection from the tumor is almost invisible. Hence, ultrasound must have

travelled in a straight line as sound is passed through the phantom and returned back from the sample's 180° end.

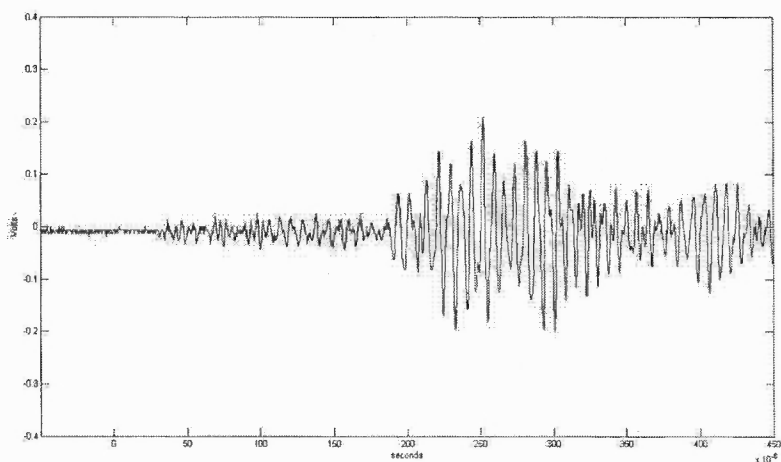


Figure 2.23 Output for echolucent phantom sample when the receiver is positioned at 45° .

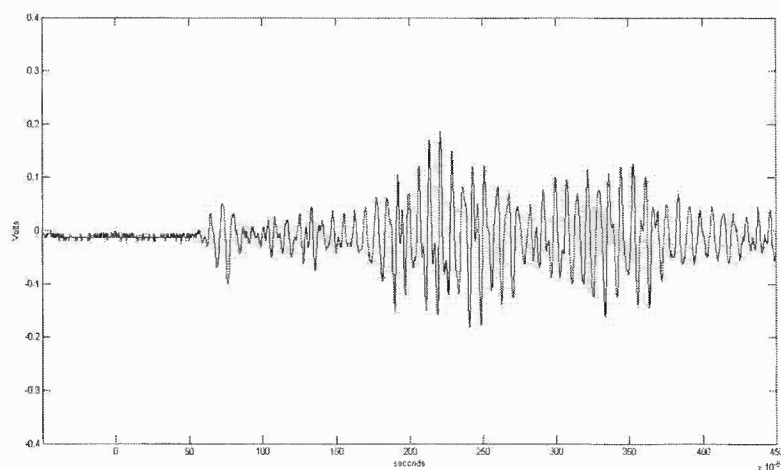


Figure 2.24 Output for echolucent phantom sample when the receiver is positioned at 90° .

In Figures 2.23 and 2.24, no significant coupling is observed and first peak arrival is the reflection of transmitted signal received on reflection from 180° opposite end.

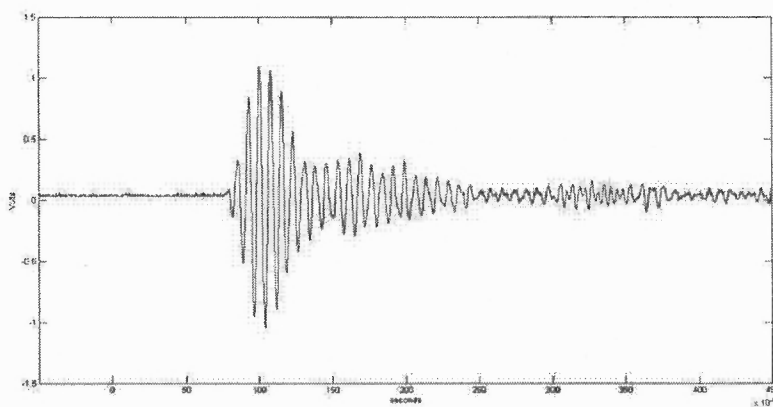


Figure 2.25 Output for echolucent phantom sample when the receiver is positioned at 135° .

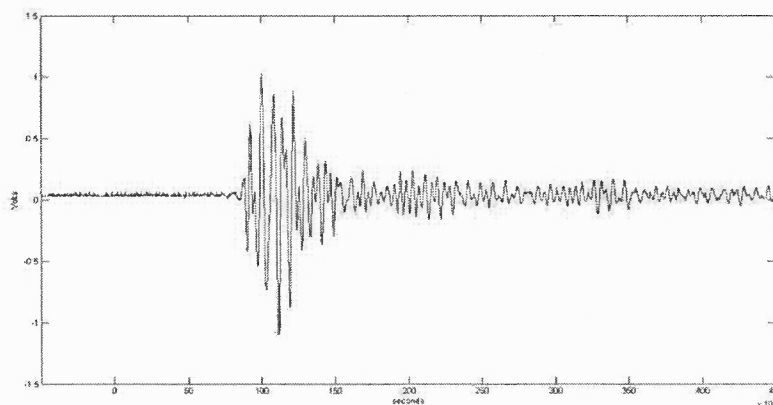


Figure 2.26 Output for echolucent phantom sample when the receiver is positioned at 180° .

In Figure 2.25, two peaks close to each other are clearly visible and in Figure 2.26, a single peak is visible, which is quite similar to the echogenic phantom sample 180° case.

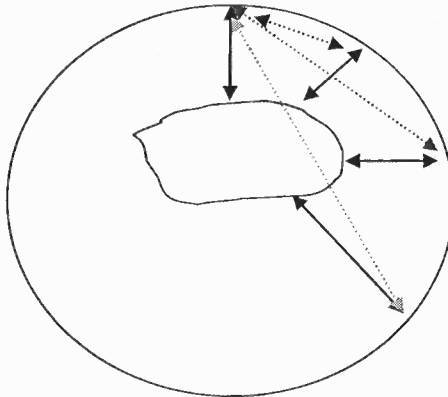
Table 2.2 shows a similar set of readings as that for echogenic samples with the values of its $\Delta T1$ and $\Delta T2$.

Table 2.2 Calculations of Approximate Lengths for Echolucent Sample

Angle (in degrees)	$\Delta T1$ (in μS)	$\Delta T2$ (in μS)	$L1 = \Delta T1 * U$	$L2 = \Delta T2 * U$
20	Immediate	183.625	NA	28.39cm
45	37.875	147.75	5.86cm	22.84cm
90	55.75	39.625	8.62cm	6.12cm
135	71	67.125	10.98cm	10.38cm
180	78.75	81.875	12.18cm	12.66cm

Similar to the echogenic phantom sample case, in the experimentation of echolucent samples, it has been observed that 45° case gives inconsistent results. In case of the echolucent sample, this scenario is not considered good for calculations.

According to (2.2), in a random scenario, the transmitter-receiver distances at various points similar to the echogenic phantom are calculated as shown in Figure 2.27.

**Figure 2.27** Computation of lengths for the echolucent sample using cosine law.

- The distance between Tx and Rx at 45° in blue is computed as 2.5cm.
- The distance between Tx and Rx at 90° in red is computed as 5.79cm.
- The distance between Tx and Rx at 135° in green is computed as 7.81cm.

The following important observations are noted in case of the echolucent sample:

- The ratio of values calculated in Table 2.2 to the corresponding values calculated above using cosine law, in case of 90° and 135° is found to be more or less equal to 1.4.
- The 25° case output shows coupling between transmitter and receiver. Hence, that scenario cannot be considered.

Difference between Echogenic Tumor Sample against Echolucent and Plain Sample:

The echolucent and echogenic tumors can be recognized mainly when the transmitter and the receiver are having the minimum possible angle without any interference between them. This is because as echogenic tumor generates an ultrasonic echo which returns to the receiver earlier than the echolucent tumor. This phenomenon can also be used to differentiate between echogenic tumor sample and plain sample.

Difference between Echolucent and Plain Sample: Echolucent sample shows similar phenomenon to that of plain sample except that it has some delay and shows multiple reflections. The 180° case is the best possible scenario to differentiate between the two as we already know the velocity of ultrasound in the soft tissue.

This analysis is further utilized to implement the system.

CHAPTER 3

SYSTEM IMPLEMENTATION

3.1 TMS320C6416T: DSK Implementation

TMS320C6416T works as center of DSP/FPGA architecture which functions as tumor location detector as well as it controls various functional aspects of system like initialization of ADC and FPGA, system timer and FPGA/DSP interface. The core of TMS320C6416T architecture consists of C6416T digital signal processor as shown in Figure 3.1, with various functional blocks

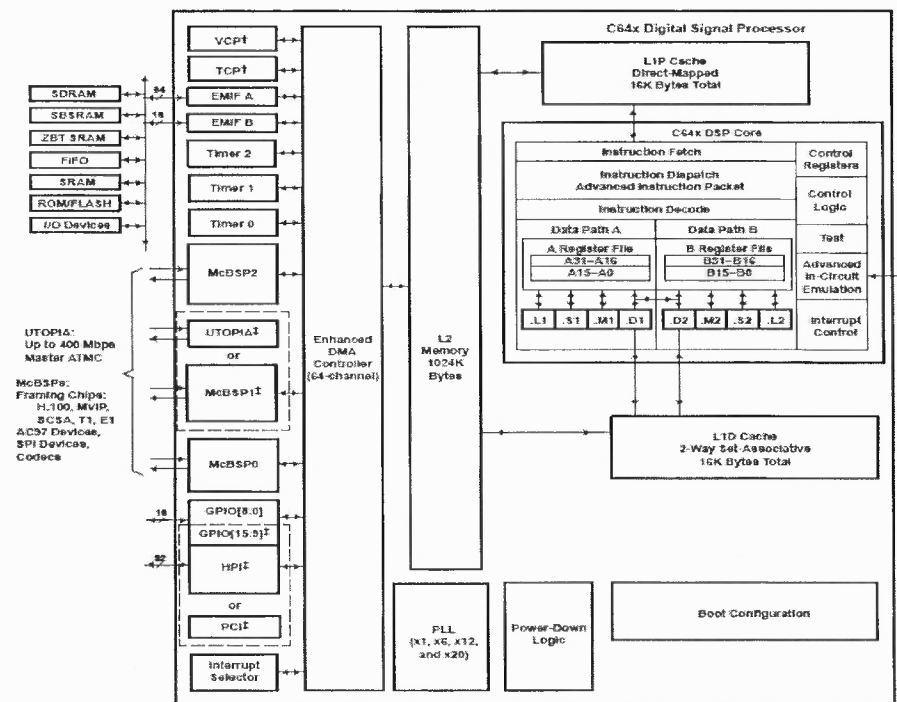


Figure 3.1 TMS320C6416T architecture.

Source: TMS320C6414T, TMS320C6415T and TMS320C6416T Fixed-Point Digital Signal Processors' Datasheet, SPRS226M, Revised April 2009.

3.1.1 TMS320C6416T Key Features

TMS320C6416T has following key features:

- 1GHz, High Performance Fixed-point processor with around 8000 MIPS speed having VelociTI.2™ Extensions to Advanced Very Long Instruction Word (VLIW) TMS320C64x DSP core [26].
- Two External Memory Interfaces (EMIFs): A 64-bit and a 16-bit interface.
- 16KB L1P direct-mapped program cache and 16KB L1D two-way set-associative data cache with 1024 KB L2 unified cache.
- Host-Port Interface (HPI) with user configurable 16 or 32-bit bus width.
- Enhanced Direct-Memory-Access (EDMA) with 64 independent channels.
- Three 32-bit General Purpose Timers.

3.1.2 Memory Map of C6416T DSK

For C64xx family, Program code and data can be placed anywhere in a unified byte addressable place. Addresses are always 32-bit wide [19]. Figure 3.2 shows the memory map of a generic C6416T processor on left with specific details of how each region is used on right.

Address	Generic 6416 Address Space	6416 DSK
0x00000000	Internal Memory	Internal Memory
0x00100000	Reserved Space or Peripheral Regs	Reserved or Peripheral
0x80000000	EMIFB CE0	CPLD
0xE4000000	EMIFB CE1	Flash
0x68000000	EMIFB CE2	
0x6C000000	EMIFB CE3	
0x20000000	EMIFA CE0	SDRAM
0x90000000	EMIFA CE1	
0xA0000000	EMIFA CE2	
0xB0000000	EMIFA CE3	Daughter Card

Figure 3.2 Memory map of C6416T DSK.

Source: TMS320C6416T DSK Technical Reference, 508035-0001 Rev. A, November 2004
http://c6000.spectrumdigital.com/dsk6416/V3/docs/dsk6416_TechRef.pdf

The internal memory sits at the beginning of the address space and portions of memory can be remapped in software as L2 cache. Each EMIF has 4 separate addressable regions called chip enable spaces (CE0-CE3). The SDRAM occupies CE0 of EMIFA while CPLD and Flash are mapped to CE0 and CE1 of EMIFB respectively. The daughter cards use CE2 and CE3 of EMIFA [19]. In our design, EMIFA is used to map Dalanco Spry DSK Helper 1 daughter card. The chip enable region CE2 (0xA0000000 – 0FFFFFFF) is allocated for the communication with onboard daughter card components. The memory connector on DSP provides access to DSP's asynchronous EMIF signals to interface memories and memory-mapped devices with byte addressing on 32-bit boundaries.

The TMS320C6416T DSK board provides a glueless interface to asynchronous memories like SRAM and EPROM and synchronous memories like SDRAM, SBRAM, ZBT RAM and FIFO. The total addressable external memory space is 1280 MB [26].

3.1.3 Timers

The three timers of C6416 have two signaling modes and can be clocked by an internal or external source with timer clock input (TINP) and timer clock output (TOUT) pins. The timer can signal an ADC to start conversion or trigger DMA controller to begin data transfer with an internal clock while using an external clock, the timer can count external events and interrupts the CPU after a specified number of events which means it acts as a counter [27]. In this design, timer_1 signals an external ADC, DAC and the FPGA (Xilinx Spartan IIE) on the Dalanco Spry's DSK Helper 1 board. Since the LTP pulse width is of 3.75 μ S, timer_1 runs at 62.5MHz frequency as C64x DSPs use CPU clock / 8 as an internal clock source [27].

$$f_t = \frac{clk}{2 * Cnt} \quad (3.1)$$

where,

f_t = timer frequency

clk = internal clock frequency

cnt = timer period register contents

This 62.5 MHz clock is provided to FPGA which divides it further for ADC (1MHZ) and other modules using a clock divider. Appendix A describes more about timer setup using its timer control register.

The system uses timer_1 that is configured through Code Composer Studio (CCS). First the counter is disabled and then it is configured by setting TOUT1 as output and enabling the clock mode and internal clock source. Then we set the number of clock cycles to be counted as '1' and then the clock is initialized.

3.1.4 Interrupts

C6416 DSP comes across various external asynchronous events including real-time data input/output. In DSP environments, interrupts are always preferred over CPU polling as in polling, CPU continuously checks whether data to be transmitted or received is ready or not, making it less efficient.

The C64x DSP core has 16 prioritized interrupts of which first four (INT_00 – INT03) are non-maskable and fixed with INT_00 being of the highest priority. INT_04 to INT_15 are maskable and can be programmed by modifying the selector values in MUXH (0x019C0000) and MUXL (0x019C0004) [26].

Interrupt Servicing: Interrupt Service Table (IST) shown in Figure 3.3 is used when an interrupt processing begins. Within each location is a Fetch Point (FP) associated with each interrupt. Table contains 16 FPs, each with eight instructions. Each interrupt has a corresponding offset address of an interrupt service routine. Each address is incremented by 20_H (32 bytes) [28].

Interrupt	Offset
RESET	000h
NMI	020h
Reserved	040h
Reserved	060h
INT4	080h
INT5	0A0h
INT6	0C0h
INT7	0E0h
INT8	100h
INT9	120h
INT10	140h
INT11	160h
INT12	180h
INT13	1A0h
INT14	1C0h
INT15	1E0h

Figure 3.3 Interrupt Service Table for C6416.

Source: Digital signal processing and applications with C6713 and C6416 DSK, Rulph Chassaing
A John Wiley & Sons Inc. Publication, January 2005.

This design implements INT_04 as hardware interrupt that indicates the completion of a task assigned to ADC on DSK Helper 1 daughter card.

3.1.5 Programming with Code Composer Studio (CCS)

This work uses C programming with CCS among other programming languages supported by it like an assembly language. CCS provides the optimization of code while compiling it into assembly.

When the program starts its execution from the 'main' function, interrupt flags are set and CCS loads context into FPGA through shared memory locations and then ADC is initialized. Then DSP waits in an infinite loop for INT_04 to occur. At the end of pre-defined sampling time, ADC module in FPGA acquires data from THS1206 and triggers INT_04 of DSP to send the data. This data is a 12-bit unsigned number and has DC reference level of 0x800 or +2.7V. This is further explained in Appendix B.

Timer Programming: Following module sets timer_1 of the system and then initializes it.

```
void timer1_start()
{
    *(unsigned volatile int *)_TIMER_CTL1_ADDR &= 0xff3f;
    *(unsigned volatile int *)_TIMER_CTL1_ADDR |= 0x301;
    *(unsigned volatile int *)_TIMER_PRD1_ADDR = 0x01;
    *(unsigned volatile int *)_TIMER_CTL1_ADDR &= 0xC0;
}
```

The first statement disables and holds the timer_1 into its current state. The second statement configures TOUT1 pin as output pin and enables the clock mode using an internal clock that is by default CPU clock/8. The third statement initializes the period of timer as 1 which means the number of clock cycles to be counted before interrupting. The last statement initializes the counter to zero and then triggers the timer_1 to start.

Interrupt programming: As stated earlier in Section 2.1.4, this system uses interrupt driven programming. An assembly source file called 'vectors.asm' is used to configure interrupt settings in the assembly language.

The vector.asm file used in this system is given below:

```
.ref _adc_isr
.ref _c_int00
.sect "vectors"
```



```

RESET_RST:  mvkl .S2 _c_int00, B0
             mvkh .S2 _c_int00, B0
             B    .S2 B0
             NOP
             NOP
.
.
.
.
INT4:       mvkl .S2 _adc_isr, B0
             mvkh .S2 _adc_isr, B0
             B    .S2 B0
             NOP
             NOP
             NOP
             NOP
.
.
.
.
INT15:      NOP
             NOP
             NOP
             NOP
             NOP
             NOP

```

This system only configures INT4 for ADC interrupt. The following points explain the interrupt configuration assembly file

- .ref command refers to ISR nomenclature in the 'main' program.
- .sect assigns the name of this section, which is used to allocate it to memory in a linker command file.
- Other interrupts are unused as ISRs for corresponding interrupts perform no operation.
- ISR of INT4 transfers the 32 bit address of _adc_isr to B0 and then it branches to the address in B0 to start interrupt processing.

EMIFA CE2 Configuration: EMIFA CE2 is a space control register (CECTL2) that has to be configured for the communication of DSP with DSK Helper 1 daughter card. The space register configuration for CECTL is explained in Appendix D. While the execution

of 'main' function, the value of EMIFA - CECTL2 is set to 0x3c0cf20 which configures the following setup [30].

- The number of clock cycles in terms of ECLKOUT1 for asynchronous read accesses is set to zero
- The interface is set as a 32-bit asynchronous interface.
- Read strobe width is set as 15 clock cycles in terms of ECLKOUT1.
- The minimum turnaround time is set as 3 clock cycles.
- Read setup width or setup time of AOE_N before ARE_N falls is set as 0 clock cycles.
- Write hold width as well as hold time of AOE_N after ARE_N rising is set to 0 clock cycle.
- Write strobe width is also set as 15 clock cycles.
- Write setup width and setup time of AOE_N before ARE_N falls for asynchronous read accesses is set as 0 clock cycle.

Loading FPGA Configuration: Function AVR32_LoadVir() loads the configuration data into FPGA through 'main' program. Table 3.1 lists the variables relevant to FPGA configuration with their significance.

Table 3.1 Variables Associated with FPGA Configuration

Variable	Significance	Initial Value
VIRTEX_PROG	It stores the address for control bits PROGHI and PROGLOW	0xa0000008 _H
VIRTEX_WRCS	It stores the address for control bits of chip select and write enable.	0xa0000004 _H
VIRTEX_MEM	The data from the FPGA file is written to SDRAM starting from this location	0x80500000 _H
VIRTEX_SIZE	FPGA memory size equivalent to 300K. Data greater than this size will be discarded	234456
VIRTEX_ADDR	Address to send configuration data to FPGA	0xa0000000 _H

The AVR32_LoadVir() can be described as follows:

1. At the beginning, CS, PROG and WRITE bits are kept disabled. Then the status of VIRTEX_STAT is checked. If both bits are high, the execution goes to next line or waits till both are high.
2. To write data into FPGA, active low chip select and write enabled are configured at address pointed by VIRTEX_WRCs.
3. DSP starts writing from the address specified by VIRTEX_MEM which is in SDRAM to the address pointed by VIRTEX_ADDR. This data transfer takes place in 'double word' format with first eight bits go into data lines 31st to 24th and then these bits are restructured. This restructuring of data takes place in to_vir() function.

```
d31 = ( 1 & inword ) ? BIT31 : 0; //Virtex d0/din LSB
```

```
.
```

```
d0 = ( 0x80 & inword ) ? BIT0 : 0; //Virtex d7
```

```
output = d31|d23|d22|d17|d11|d6|d5|d0;
```

When the transfer is completed the CS and WRITE signals are deactivated by pulling them 'high' and then it switches back control to 'main'.

ADC (THS1206) Programming: The programming flow skips the default configuration of ADC and configures it as per the system requirements as shown in Figure 3.4.

First it writes 0x401 to THS1206 which sets 'Reset Bit' in CR1 and then it clears RESET by writing 0x400 to CR1. Since the system uses the user configuration mode, the program writes 0x1a0 to CR0 which configures following things [21]:

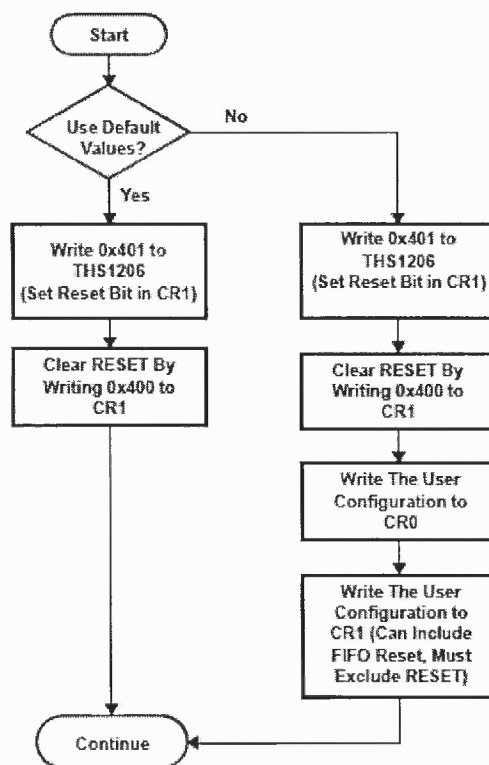


Figure 3.4 THS1206 configuration flow.

Source: THS1206 Datasheet, SLAS217H, Revised – July 2003, Texas Instruments, Inc.

- Internal voltage reference is selected.
- Continuous conversion mode is selected.
- ADC is active.
- Analog input BINP (Single ended).

Similarly CR1 is configured to 0x4d2 which means

- No Reset for ADC
- FRST flag set to '1' which means FIFO is reset.
- FIFO trigger level is set to '00' for the single channel.
- DATA_AV bits are set to '01' which means active low pulse.
- R/W (Active Low) mode is selected and RD (Active Low) disabled.
- Output value of ADC is in binary format.
- Normal Conversion mode is selected.

Project File (.pjt): This is the file that groups the entire project. It contains different source files, libraries, configuration files, and include files. When the project is built, all files in CCS projects are integrated to .out file [29] and this file is downloaded on DSP for execution.

Linker File (.cmd): Linker file is used to map different sections of memory and each project can have only one linker file [28]. The linker places various sections into appropriate memory locations where code and data resides. This file is customizable having MEMORY and SECTION directives [28].

The 'Memory' directive specifies all the memories or interfaces associated with DSP that includes ISRAM, SDRAM, and CE2 region of EMIF with its base address and length. It also defines interrupt vector locations.

The 'Section' directive allocates different code sections in different available memory spaces [28]. The mnemonics used are assembler directives that are resolved during assembling.

The linker file used in the system is as follows:

```
MEMORY
{
    vecs:      o = 0x00000000 l = 0x00000200
    ISRAM      : origin = 0x200,      len = 0x1000000
    SDRAM      : origin = 0x80500000, len = 0x400000
    CE2        : origin = 0xA0000000, len = 0x400000
}

SECTIONS
{
    "vectors" > vecs
    .text > ISRAM
    .bss > ISRAM
    .cinit > ISRAM
```

```

.const > ISRAM
.far > ISRAM
.stack > ISRAM
.cio > ISRAM
.sysmem > ISRAM
.mw_isrambuff > ISRAM
}

```

FPGA File (.out): Xilinx ISE generates .hex file that is further converted into .out file.

This file is loaded into FPGA via DSP and hence, it becomes necessary to load it into DSP SDRAM using 'Load Data' option in Code Composer Studio IDE.

Execution of Main Program: The initial program for DSK provided by the manufacturer is modified to suit the requirement of this design. The current version of the program carries out the following things in sequence:

- The initialization process is carried out while clearing all previous interrupts and setting the interrupt again. The CSR (Control Status Register) is configured to 0x100 to disable all interrupts. Then IER (Interrupt Enable Register) is configured to 0x0001 to disable all interrupts except NMI. ICR (Interrupt Clear Register) is configured to 0xffff to clear all the pending interrupts [28].
- Then it sets EMIFA CE2 space control register as explained in before.
- It initializes addresses for DAC1 and DAC2.
- The program initializes cache and halts the timers, timer_0 and timer_1 but holds its value.
- On halting the timers, the program executes AVR32_LoadVir() function for loading configuration data into FPGA for its functioning and again it restarts timer_0 with clock cycles to be counted as 20_H.
- Then it initializes ADC (THS1206). In this system configuration, the program skips ADC's default configuration and configure it for this system's purpose as explained before.
- The program then enables timer_1 and interrupt INT4 for timer_1 using the following code as well as it maintains a count with a variable called 'countadc' each time timer_1 interrupt occurs.
- This program works in an infinite loop as explained before.

- Whenever an interrupt is generated by ADC, the demodulated signal is fetched into DSP for further data processing.

3.1.6 ADC Interrupt Processing

The interrupt, which is generated by ADC, is processed by dividing into two tasks:

Filtering: Filtering is necessary to remove the high frequency noise components from the demodulated signal received at DSP. This makes simple the task of detecting maxima for the estimation of size of tumor. The Interrupt Service Routine ‘_adc_isr’ implements ‘Simple Moving Average Filter’ for this purpose.

The demodulated data is used as input to moving average filter designed in Matlab and processed for improved results with window = 16. This removes the most of high frequency noise while giving clear maxima. The design is further implemented with DSP using FIFO. On each ADC interrupt, 12-bit data is captured from QAM, converted into signed and stored in an array of size 16. When the array is filled, all the values are added and then divided by 16 to obtain desired results. Figure 3.5 shows demodulated envelop at DAC2 output while Figure 3.6 shows the desired improvement after simple moving average filtering.

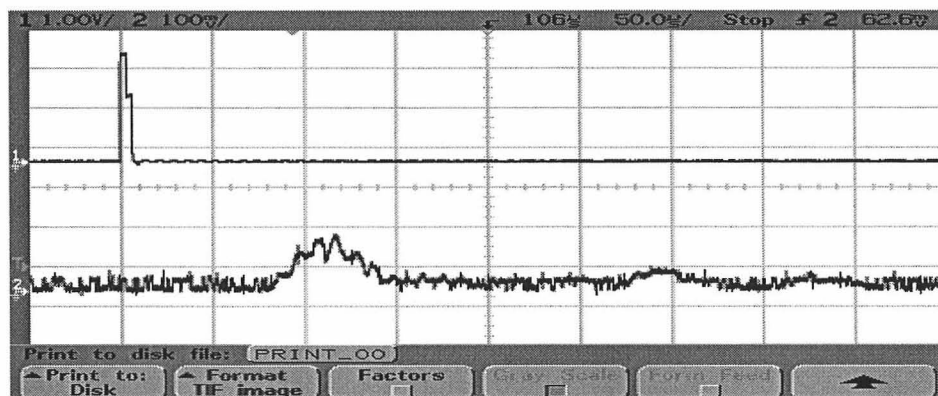


Figure 3.5 Demodulated envelop at DAC2 output.

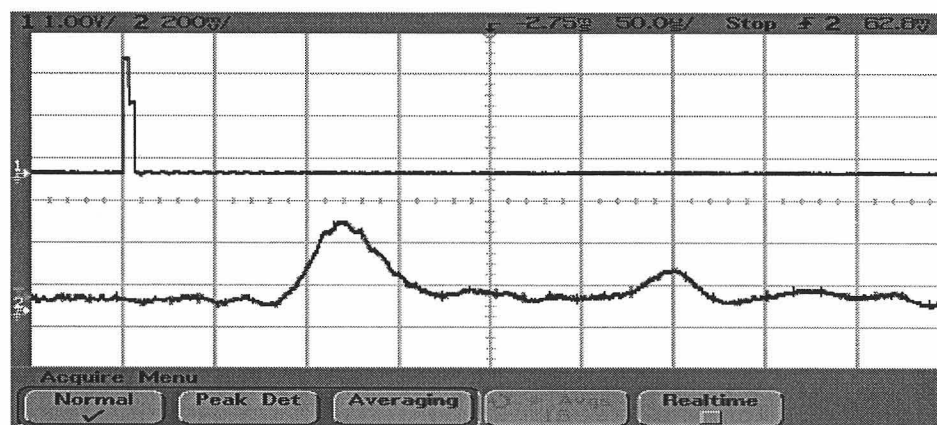


Figure 3.6 Demodulated envelope after simple moving average filtering at DAC2 output.

Locating Tumor: DSP detects first two maxima to calculate the location of a tumor. The QAM demodulated data is filtered first and then an algorithm is implemented to detect first two peak locations of the envelope. These peak locations help to carry out the analysis of different parameters of tumor samples.

QAM demodulator's current output is compared to its previous value to determine which is greater and it is stored as maximum. This comparison is done until the envelope reaches its first maximum. A counter is incremented at each interrupt which occurs at $1\mu\text{S}$ till this time and it is displayed as 'max1' in the watch-window of output.

For the second maximum, it is considered as lesser than the first maximum and secondly, it has to avoid the detection of any point as a maximum from the first envelope. The algorithm should detect the maximum on the rising curve and display it as 'max2' when found. One sample output in the watch-window of Code Composer Studio is as shown in Figure 3.7.

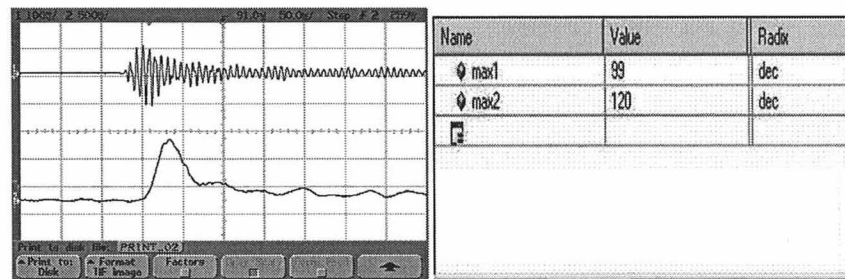


Figure 3.7 Sample output and watch window output in Code Composer Studio.

3.2 Xilinx Spartan-IIE: FPGA Implementation

The system uses Xilinx Spartan-IIE FPGA with device ID XC2S300E. This device has 6,912 logic cells with a typical system gate count ranging from 93,000 to 300,000 which provides 32 x 48 (R x C) configurable logic blocks (CLBs). Table 3.2 provides more details regarding Spartan IIE family [20].

Table 3.2 Xilinx Spartan-IIE Family

Device	Logic Cells	Typical System Gate Range (Logic and RAM)	CLB Array (R x C)	Total CLBs	Maximum Available User I/O ⁽¹⁾	Maximum Differential I/O Pairs	Distributed RAM Bits	Block RAM Bits
XC2S50E	1,728	23,000 - 50,000	16 x 24	384	182	83	24,576	32K
XC2S100E	2,700	37,000 - 100,000	20 x 30	600	202	86	38,400	40K
XC2S150E	3,888	52,000 - 150,000	24 x 36	864	265	114	55,296	48K
XC2S200E	5,292	71,000 - 200,000	28 x 42	1,176	289	120	75,264	56K
XC2S300E	6,912	93,000 - 300,000	32 x 48	1,536	329	120	98,304	64K
XC2S400E	10,800	145,000 - 400,000	40 x 60	2,400	410	172	153,600	160K
XC2S600E	15,552	210,000 - 600,000	48 x 72	3,456	514	205	221,184	288K

Source: Spartan-IIE FPGA Family Datasheet, DS077, June 2008.

There are different packaging options available with the FPGA but the system implemented here utilizes PQG208 packaging having 146 user available I/Os [20].

3.2.1 Programming Steps

The programming development cycle is divided into following steps:

1. The architecture is first designed, examined and optimized with a high level simulation of the complete system including transfer functions. The transfer function is estimated using the corresponding datasheets of devices.
2. The different sub-blocks of the system are translated into Matlab/Simulink for hardware implementation.
3. The validation of the corresponding VHDL program is done with simulator output and the experimental hardware output is also verified.
4. An algorithm is designed to operate on the captured demodulated data in DSP memory to determine the tumor location.
5. The system is designed and implemented on DSP with calibration for noise removal.

3.2.2 Simulation of Quadrature Amplitude Demodulation

The Quadrature Amplitude Demodulator (QAM, also known as 'I and Q Modulation') is designed in Simulink first to determine the parameters of the different components. This demodulation technique is required for receiving back the baseband signal from the high frequency modulated carrier signal as well as removing high frequency noise components.

The QAM demodulator needs in-phase (I) and quadrature-phase (Q) carrier components, high frequency noise filters, multipliers, an adder and a square-root calculator. Figure 3.8 shows Simulink design of the demodulator. The transducer transfer function is first determined which modulates the pulse sequence. It is given by

$$G(s) = \frac{\alpha s}{s^2 + 2\zeta\omega_n s + \omega_n^2}$$

where,

$$\zeta = 0.0858;$$

$$\alpha = 5000;$$

$$\omega_n = 839390.$$

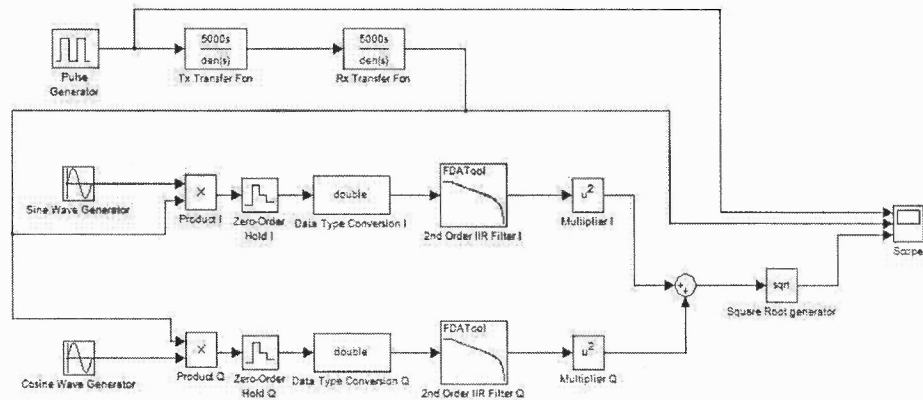


Figure 3.8 QAM Simulink design.

The output from transducer transfer function blocks is synchronously multiplied with two 90° out-of- phase carriers with carrier frequency 150 KHz. The filter cut-off frequency is set at 75 KHz, half the carrier frequency. Then both these signals are squared and added together before calculating square-root. Figure 3.9 displays the simulated oscilloscope outputs of input signal, receiver signal and demodulated signal.

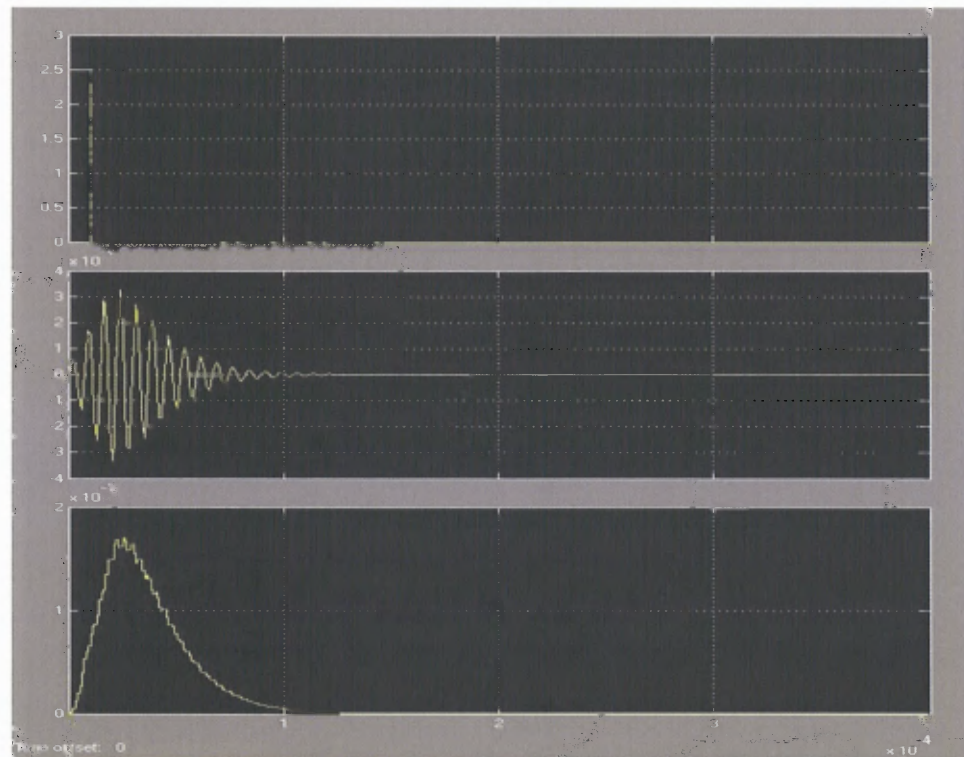


Figure 3.9 Simulated outputs for input, receiver and demodulated signals.

3.2.3 FPGA Programming

FPGA programming for the system is done using VHDL semantics in Xilinx ISE v6.1.

The approach used in design is called Medium Scale Integration (MSI) approach. The design flow is given as follows:

- The functions are divided into adders, multipliers and flip-flops.
- Each function is then written into VHDL in Xilinx ISE v6.1 and its simulation output is checked in ModelSim simulator.
- The different blocks created together to form larger blocks like sine wave generator, square root generator etc.
- These results are again simulated in ModelSim simulator.
- These functional blocks are created and then port-mapped to form the functions like QAM etc.

Low Transient Pulse Generation: LTP drive signal is generated using FPGA to achieve high temporal resolution and better detection performance [8]. The timer is used at its maximum frequency of 62.5MHz to achieve high temporal resolution. FPGA is configured as a 12-bit module preloaded with the corresponding binary value of each pulse to drive a DAC converter. The values are output through a circular buffer.

The register transfer level (RTL) implementation of Low Transient Pulse (LTP) in DAC module of FPGA is shown in Figure 3.10. The counter register CNT is incremented at each clock and is checked against less than or equal to the function. If the value is less than 202 then it outputs 2.6V but if value is greater than 202 but less than value in B then it outputs 1.6V. XWEn and Chip select for DAC are ORed to get write control bit for DAC.

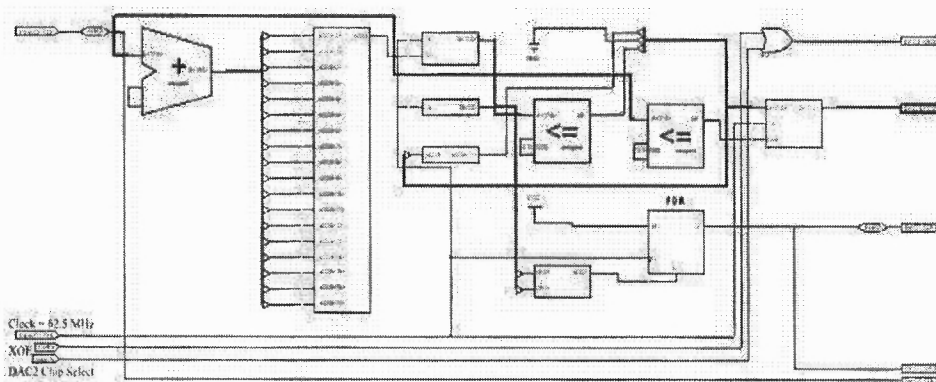


Figure 3.10 Register transfer level diagram for Low Transient Pulse.

Clock Divider: As the design requires different modules running at different clock speed, a clock divider is implemented to run LTP and DAC at higher frequency while ADC and other modules at a relatively lower speed.

Figure 3.11 explains register transfer level (RTL) of the clock divider. The counter increments at each clock edge and is checked against count 31. If the incremented value is less than 31 then input of FDR block is transferred to clk_out signal which is 'high' otherwise when counter reaches 31, the output of 'less than or equal to' block becomes 1 which eventually resets FDR block giving 'low' as clk_out output. Thus, 62.5 MHz clock input gives the output of clock frequency using (3.1) as follows:

$$\text{Timer Frequency} = 62.5 \text{ MHz} / (2 * 31) = 1 \text{ MHz}$$

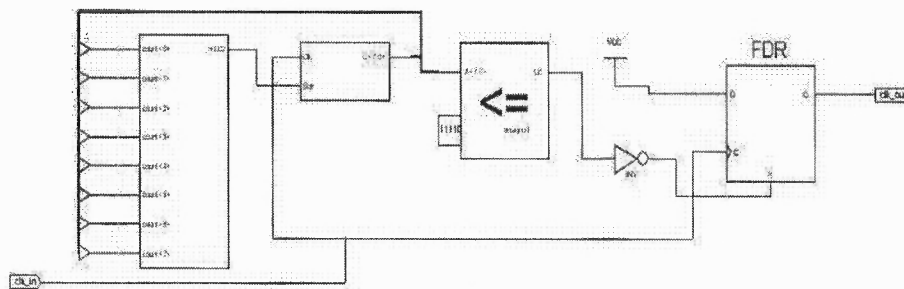


Figure 3.11 Register transfer level diagram for clock divider.

FPGA and DSP Interface with Address Decoding: The C6416T DSP has two external memory interfaces (EMIF): 64-bit EMIFA that is device specific and 16-bit EMIFB. These EMIFs can be utilized to connect various daughter cards like the one used in this system. The 'Read to Write' latency for the EMIF interface is CAS latency plus 2 EMIF cycles to provide 1 turnaround cycle. This is the maximum latency required depending on TCL bits which is either 3 cycles (TCL = 0) or 4 cycles (TCL = 1) [30].

DSP requests data on its 64-bit bi-directional address/data EMIFA bus from ADC by communicating with FPGA. When it receives data, it processes it and meanwhile, it switches from read cycle to write cycle to write into DAC. The data from DSP is sent back after processing to DAC module of FPGA for conversion into analog baseband signal by sending data at addresses starting from location WRDAC1 and WRDAC2 for channels 1 and 2. The EMIF CECTL is setup with the parameters as described in Section 3.1.5.

Since ADC interrupts the DSP at every $1\mu\text{S}$, it is required to be processed within that period. The ISR for ADC interrupt processes the data and then returns to the 'main' program that in turn waits into a infinite loop for the next interrupt to occur. Table 3.3 explains the memory interfacing between the DSP and FPGA.

Table 3.3 Memory Interface between DSP and FPGA

Hex Address	Identifier	Significance
0xa0000000 – 0xaFFFFFFF	EMIFA	Daughter card Memory Space
0x01800010	EMIFA_CE2	Address of EMIF Space Control Register CECTL2
0xa0080000	CSADC	ADC configuration address
0xa0080004	WRDAC1	Address to send data from DSP to DAC1 of daughter card
0xa008000C	WRDAC2	Address to send data from DSP to DAC2 of daughter card

Source: TMS320C6416T DSK Technical Reference, Spectrum Digital Inc., 2004; TMS320C6000 DSP EMIF Reference Guide, SPRU266E, Texas Instruments Inc., April 2008.

Table 3.3 clearly explains that FPGA is interfaced with DSP using DSP's External Memory Interface (EMIFA). Figure 3.12 explains register transfer logic (RTL) developed for decoding address and control lines for FPGA from addresses and control signal generated by DSP. Lower 16 lines of the 32-bit address bus are used for FPGA. These address line sequences decide the address space for which the data is. Since the double-word addressing mode is used, bits 0 and 1 are not used for decoding. They are used as chip-select to activate a particular on-board memory chip.

During the initialization cycle, data bus used to send data from DSP to FPGA for ADC configuration. Figure 3.12 shows the 15th, 14th, 13th address bits are ORed with write- enable and decode signal to generate *rdN* which decides the direction of data on bi-directional data bus. If it is low, data is received into DSP otherwise it is used to send the data to ADC.

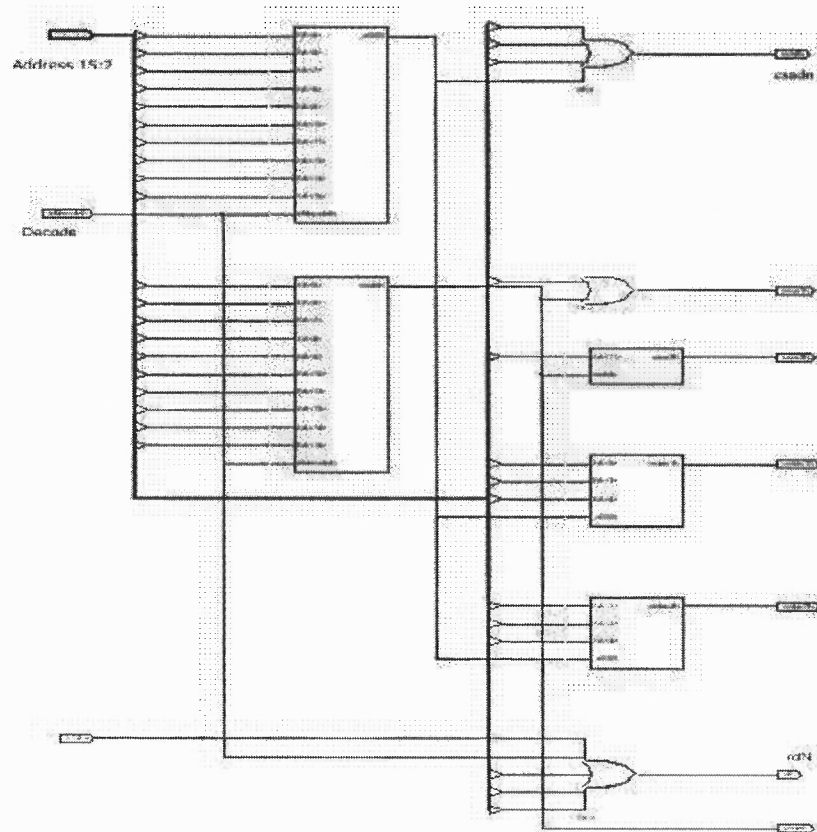


Figure 3.12 Register transfer level for decoder to interface FPGA and DSP.

To transfer data between ADC and DSP, the address bits 15-2 are ORed along with the decoder bit to check whether all of them are low or not to generate active low *csadN* which enables writing to the address of CSADC (0xa0080000), the ADC configuration registers.

To transfer data from DSP to DAC1, an active low control signal *csdac1N* is generated by doing an OR operation on all address bits (15-2) with bit 2 negated which redirects to the address of WRDAC1 (0xa0080004). It is shown in Figure 3.13. Similarly, to transfer data from DSP to DAC2, an active low control signal *csdac2N* is generated by doing the similar OR operation on all address bits (15-2) with bit 3 and bit 2 negated

which redirects to the address of WRDAC2 (0xa008000C) which is shown in Figure 3.14.

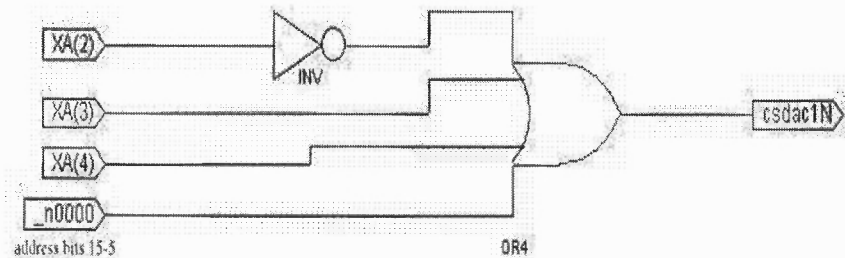


Figure 3.13 Register transfer level for generation of control signal for DAC1.

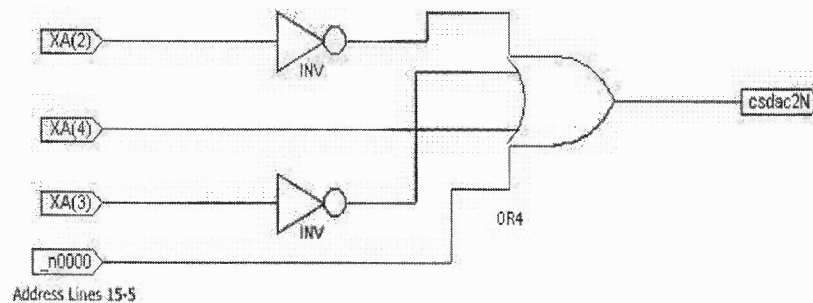


Figure 3.14 Register transfer level for generation of control signal for DAC2.

Analog to Digital Converter: This module establishes the communication between ADC on the daughter card and DSP. The module transfers data into ADC as well as configures ADC during system initialization.

Figure 3.15 describes the register transfer level of an ADC module. *toadclk* is clock signal for ADC which runs at 1 MHz. *XWE* and *csad* are ORed to generate control signal *ADRW* which when low, transfers data from DSP to ADC for initialization, otherwise, DSP receives 12-bit sampled data from ADC on data bus.

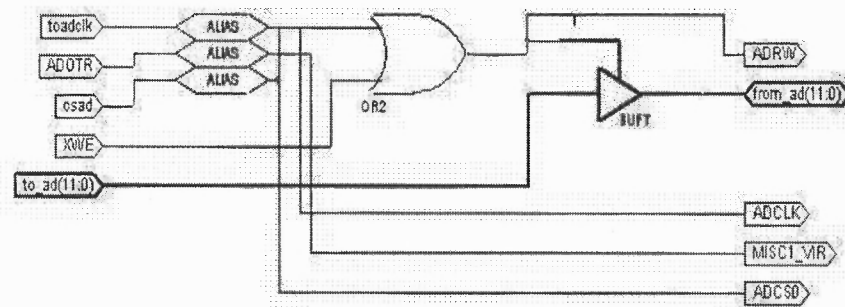


Figure 3.15 Register transfer level for analog-digital converter.

3.2.4 Quadrature Amplitude Demodulator (QAM)

The Simulink design of QAM, explained in Figure 3.8, is implemented on FPGA during which each component is designed in VHDL and then mapped into a schematic diagram in Xilinx ECS. In this thesis, the design of each component is described in brief and in order of occurrence.

Sine and Cosine Carrier Generator: A circular buffer with fixed point values derived from Matlab is constructed for 150 KHz sine as well as cosine wave which is sampled at 1 MHz. At each clock pulse, a buffer generates a value and increments the pointer. It can be mathematically proved that the output values from buffer are repetitive after three periods or 20 samples of sine and cosine waves as

$$\frac{1\text{MHz}}{150\text{KHz}} * 3 = 20$$

The buffer is designed in the VHDL program which, on each clock, increments the pointer and outputs the corresponding value from a look-up table. A 6-bit data length is found to be sufficient in which the most significant bit indicates the sign of the value and remaining 5 bits give the amplitude of the wave within the range (-31, 31). The Matlab simulation for the plotted sine values is shown in Figure 3.16.

Matlab Program:

```
%fs = 1 MHz
%f = 150KHz
for i= 1:20
    y(i) = 31 * sin(2*pi*150000/1000000*i);
end
y = round(y);
```

The 20 sample outputs generated are 25, 29, 10, -18, -31, -18, 10, 29, 25, 0, -25, -29, -10, 18, 31, 18, -10, -29, -25 and 0.

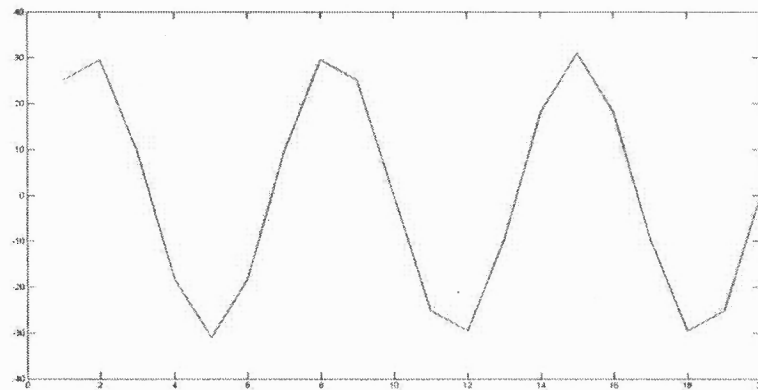


Figure 3.16 Matlab simulation of a sine wave generator.

A similar look-up table is used for a cosine wave generator and both are simulated in ModelSim simulator as shown in Figure 3.17. The decimal data is first converted to 6-bit signed binary data. The wave generator resets first and then for each clock cycle provided to its clock input, the counter is incremented and then outputs corresponding value from the look-up table starting from 25 (011001_2) for sine wave generator. When counter reaches 20th value, it starts again with first value on next clock cycle.

This block consists of a counter (B), two multiplexers (C) and two latches (D and E). At each clock cycle, counter B is incremented to derive the next integer value from A which is then used to select a corresponding value from look-up table C. The value is latched in latches D and E and on each rising edge of the clock, the latched value is output on sine and cosine output lines. The 'reset' input is used for asynchronous reset to re-initialize the counter and reset the latches.

Filter Design: Implementation of a finite impulse response filter (FIR) cannot be considered as a low pass filter for the requirement as Spartan-III FPGA has lower gate count than required to implement a higher order FIR filter. A second order Butterworth IIR filter is implemented for this purpose as it requires less number of gates. Another advantage is, as filter implemented in the forward loop, phase delays introduced by filter can be ignored. In this case, they are found to be zero.

A Direct-II form Low Pass Butterworth IIR Filter is implemented on FPGA since it requires less memory for implementation as it requires less number of delay elements [34]. While implementation, filter gain is kept at minimal so as to avoid any saturation that might occur. It can be increased further depending on input to the QAM system. Figure 3.19 shows Direct-II form transform domain IIR filter. 'X' indicates input. 'Y' indicates output and Q_n and P_n indicate poles and zeros, respectively.

The filter coefficients are scaled as they are fractional. Fixed point simulation is carried out in Matlab using 'butter()' function and compared with the Simulink model shown in Figure 3.8. The coefficients are then broken down into multiples of $(\frac{1}{2})^n$ [34].

Table 3.4 indicates the parameters defined in the Simulink with the magnitude response graph plotted.

Table 3.4 Simulink Parameters for IIR Filter

Cut-off Frequency	75 KHz
Sampling Rate	1 MHz
Filter Type	Second Order IIR Butterworth Filter
Filter Denominator	1; -1.3489; 0.51398
Filter Numerator	1; 2; 1
Filter Gain	0.041253

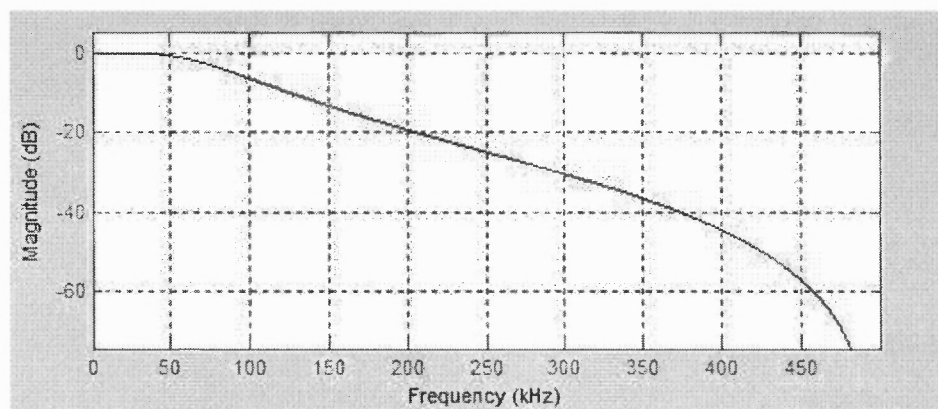


Figure 3.20 Magnitude response of IIR filter designed using parameters in Table 3.4.

The VHDL implementation of an IIR filter is done according to (3.2) to (3.6). Its Register transfer level logic is shown in Figure 3.20.

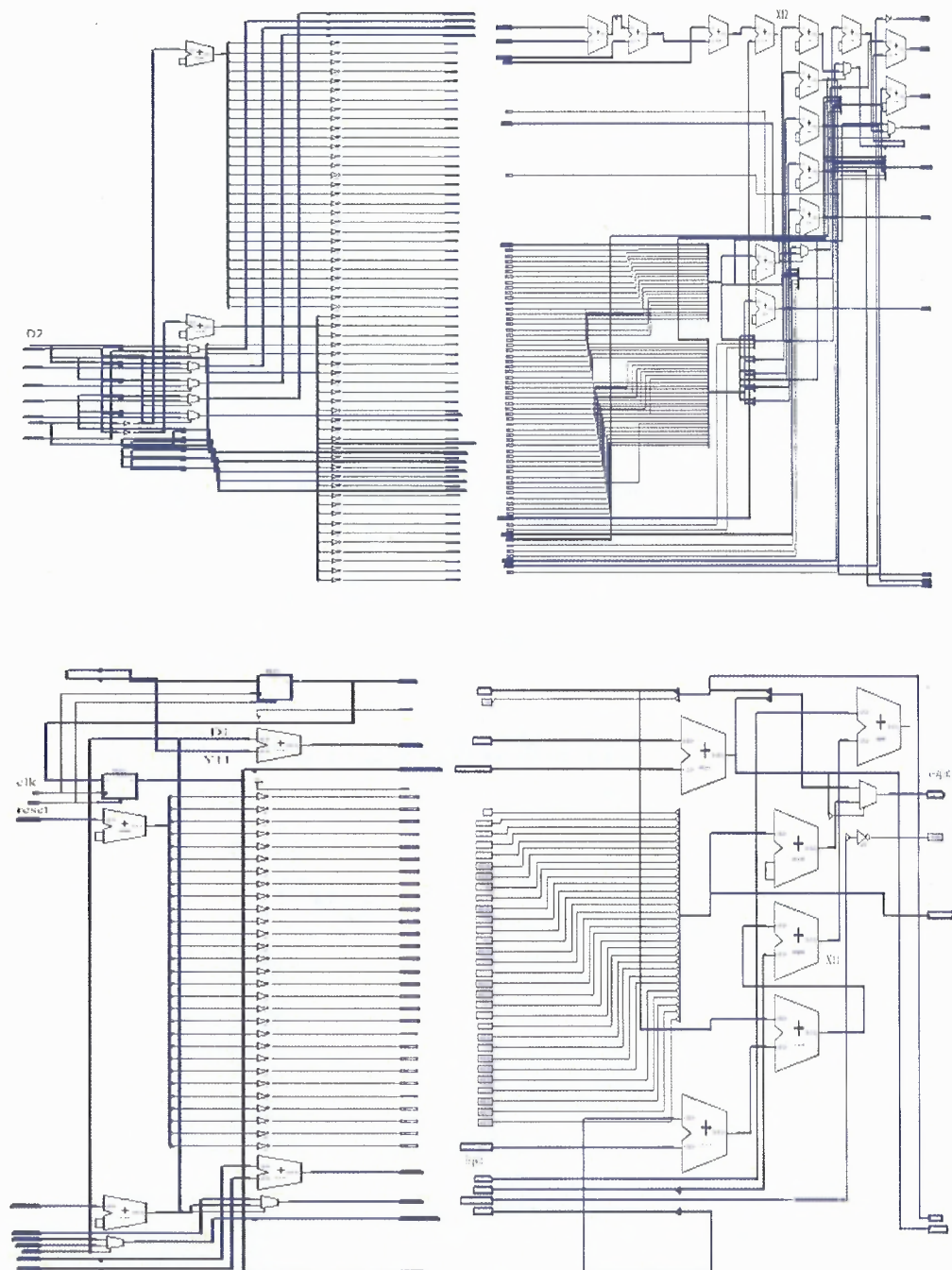


Figure 3.21 Register transfer level for IIR Filter.

Square Root Generator: The square root calculation is implemented in this work using Restoring Binary Shift-and-Subtract Square Rooting Algorithm [35]. The restoring process is achieved by a multiplexer selecting the previous remainder if the subtractor

leads to a negative result. This subtractor subtracts expression $P(i)$ from successive remainder $R(i-1)$ and final result is the concatenation of sign bits of Q . $P(i)$ can be calculated using the following equation.

$$P(i) = (4 * Q(n-i) + 1) * 2^{2(n-i)} \quad (3.7)$$

In an input vector of length $2n$, the algorithm calculates the square root of length n and is the value in $Q(-1)$ in the n^{th} step. The sub-blocks of adder, shifter and multiplexers are designed and optimized to perform the square root operation.

The simulation results of the square root generator are shown in Figure 3.22.

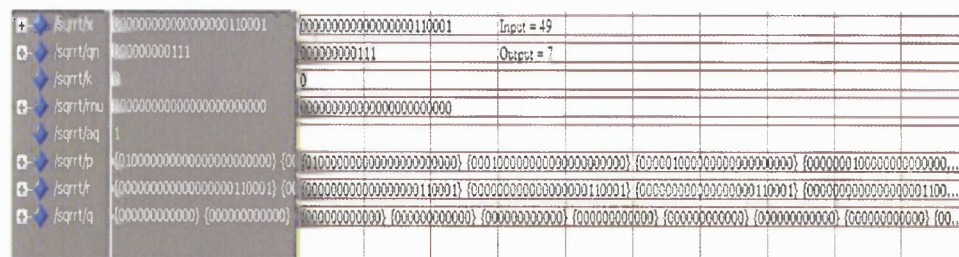


Figure 3.22 Simulation of the square root generator.

Figure 3.23 shows register transfer level logic for the square root generator. Figure 3.23 shows the top level module that has 11 sub-modules and a subtractor. 11 sub-modules reflect $n-1$ (which is, in this case, 11) steps in the algorithm.

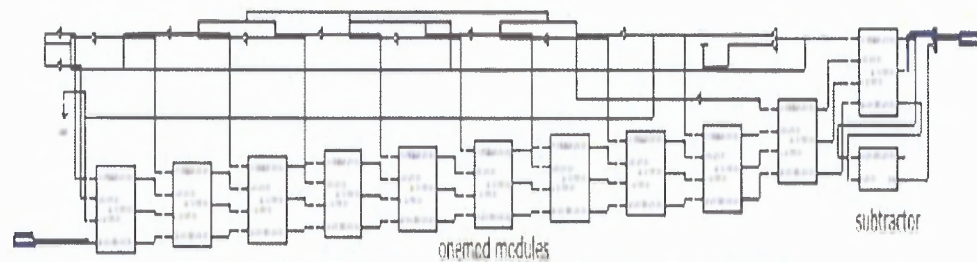


Figure 3.23 RTL of high level module for square root generator.

The final remainder is ignored as it is not considered further. Figure 3.24 shows the register transfer level logic of an onemod module with a single step of the algorithm. It consists of a subtractor and cali block. $P(i)$ and $R(i-1)$ are the inputs to this onemod block. Figure 3.25 consists of a subtractor followed by a multiplexer that decides which value to output. The inverse of the most significant bit of the output from the adder is used as input to shifter.

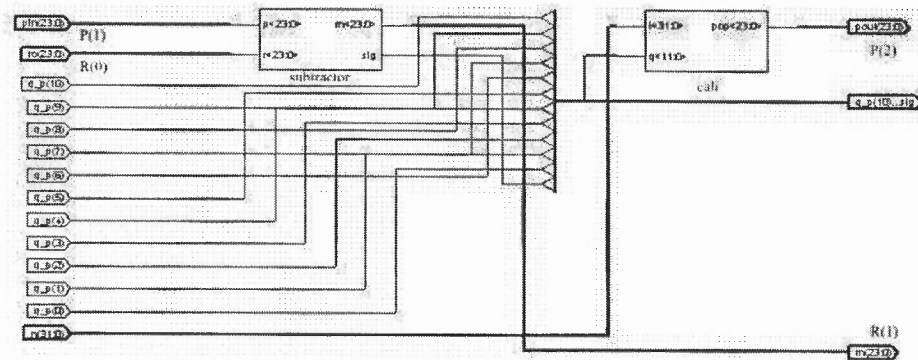


Figure 3.24 Register transfer level of Onemod module for square-root generation.

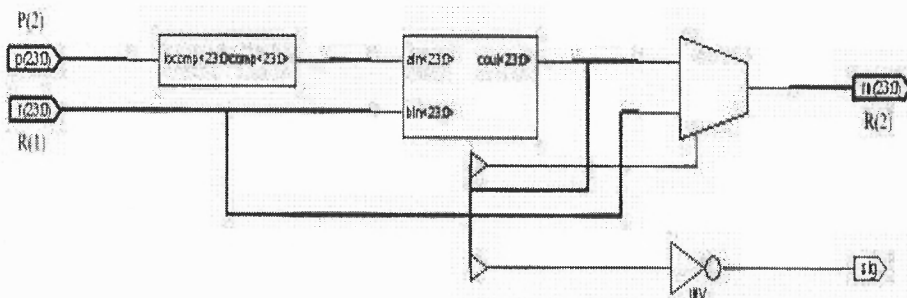


Figure 3.25 Register transfer level of a Subtractor module.

The complete module of QAM is then port-mapped according to Simulink model as shown in Figure 3.8. Figure 3.26 describes complete QAM schematic.

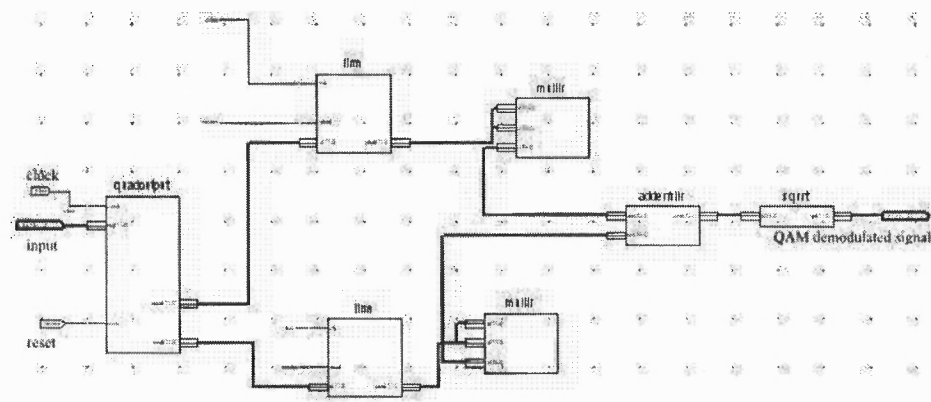


Figure 3.26 Schematic of QAM demodulator.**Frequency Spectrum Analysis of QAM:** QAM demodulation is performed at a high frequency on FPGA environment while DAC performs low frequency envelop detection tasks.

The receiver signal has frequency of 150 KHz which is sampled at 1MHz for QAM. If these samples are directly sent to DSP for calculations without QAM demodulation, then an interrupt occurs for every $1\mu\text{S}$ at ADC input that triggers an execution of the algorithm. This algorithm is complex enough to finish its execution in $1\mu\text{S}$ on DSP.

This is overcome by QAM demodulation in FPGA. The QAM demodulation shifts the frequency spectrum of the received signal more towards zero giving almost four times more time for the execution of instructions on DSP so that a complex algorithm can finish its execution. Figure 3.27 and 3.28 show the frequency spectrum of the received signal while Figure 3.29 and 3.30 show QAM demodulated output and its frequency spectrum.

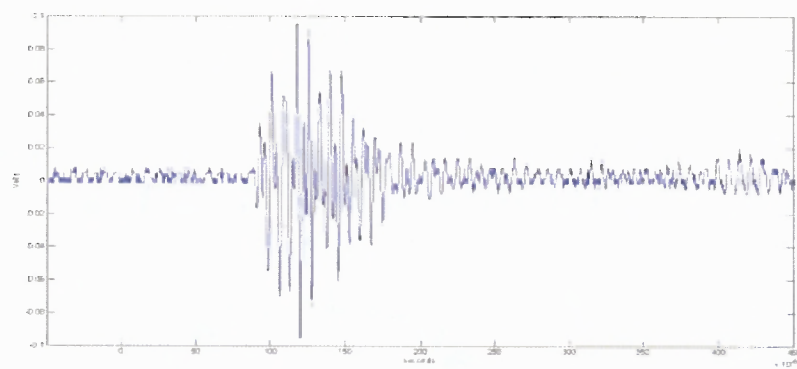


Figure 3.27 Output of receiver.

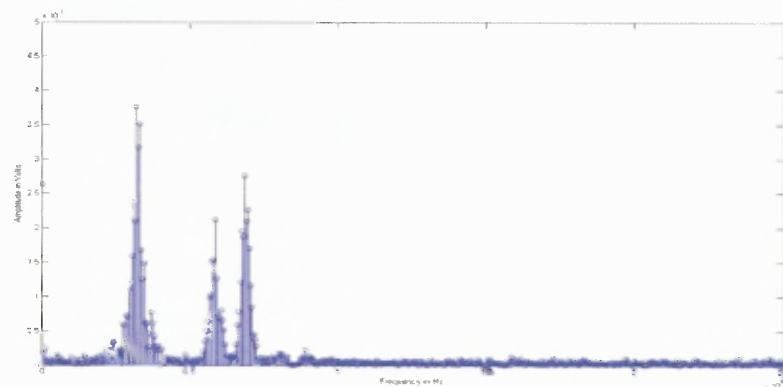


Figure 3.28 FFT of output of receiver.

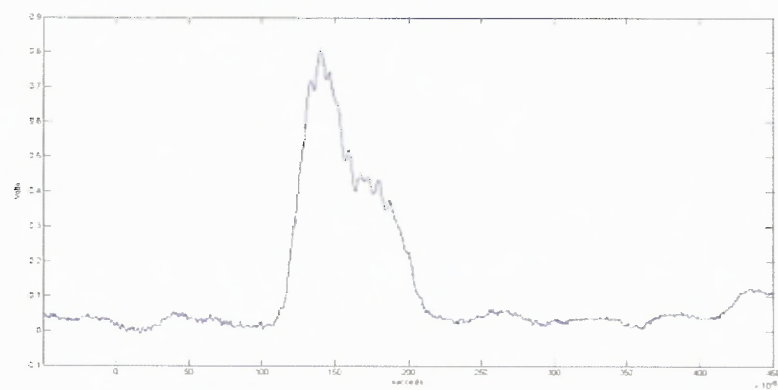


Figure 3.29 Demodulated output of QAM.

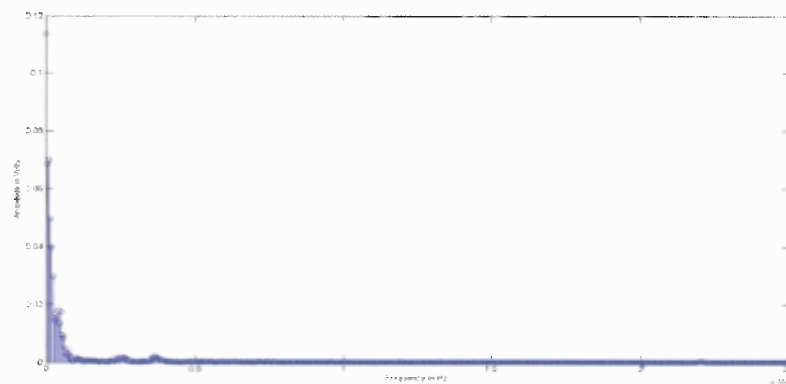


Figure 3.30 FFT of demodulated output of QAM.

Unsigned to Signed and Signed to Unsigned Converter: The data coming from ADC is unsigned and first, it has to be converted into signed before sending to QAM. Similarly, data coming out from QAM is to be converted into unsigned. This is done by *unsignedtosigned* and *signedtounsigned* modules.

3.2.5 Entire FPGA System

Figure 3.31 shows the interfacing of different modules of FPGA to carry out intended tasks of the generation of low transient pulse (LTP) and the demodulation of signal.

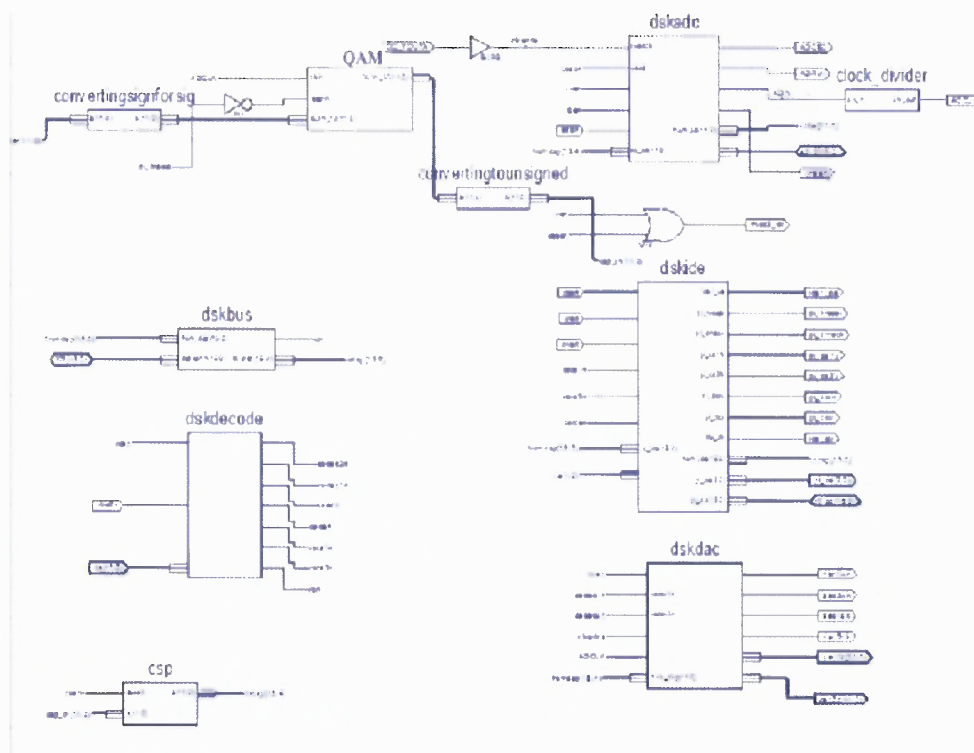


Figure 3.31 Interconnections of FPGA modules.

ADC sends the sampled data to the ADC module through bidirectional AD I/O pins. The data is first converted to signed and then sent to QAM for the detection of baseband signal. Once the signal is detected, it's sent back to DSP using *dsdbus* module after converting into the unsigned first. DSP carries out moving average filtering and peak detection. This filtered data is sent back to *dskdac* through a bidirectional XD I/O port which subsequently sends the data to DAC2. This *dskdac* is also responsible for low transient pulse synthesis that is output on DAC1. The LTP generation takes place at a very high frequency of 62.5 MHz and ADC is run at 1 MHz. Hence, the *clock_divider* module is used for this purpose.

CHAPTER 4

EXPERIMENTS AND RESULTS

4.1 Execution Flow

The flow of the experimentation for tumor detection in ultrasound samples can be explained as follows.

1. LTP is synthesized on the DAC1 module of the daughter card with the maximum possible resolution of 62.5 MHz. It has two pulses of $A_1 = 2.5V$ and $A_2 = 1.5V$ with each having 3.27ns width. The LTP output has 400 Hz frequency. This LTP drives Physical Acoustics' 150 KHz R15 α transducer.
2. The transducer at the receiver end captures the signal from the test sample that is further driven by Physical Acoustics' Wide bandwidth AE amplifier.
3. This signal is passed through THS1206 ADC that samples the signal at 1 MHz and generates 12-bit digital output which is further given as input to the QAM demodulation module.
4. In the QAM demodulation system, the 12-bit ADC output is first multiplied with 6-bit signed sine/cosine wave generator output to produce in-phase and quadrature-phase (I and Q) components with sample rate 1 MHz to produce 18-bit sampled vector streams. These 18-bit vectors are passed through two second order Butterworth IIR filters, one for in-phase and another for quadrature phase. Then both of them are squared and added together to produce a 25-bit vector. A square-root generator produces a 12-bit square root vector from this 25-bit vector which is the envelope of

the receiver signal.

5. This demodulated data is sent back to DSP through EMIF when a hardware interrupt, setup at 1 MHz, occurs.
6. DSP processes data to find out peaks to determine the location of a tumor in the tumor sample. This is done by removing high frequency noise using a simple moving average filter with window size = 16.
7. To find first two maxima, a first maximum is detected when the first peak is observed but for the second maximum, leading as well as falling edge is also checked. The maxima will only be reported if it is on rising edge. These maxima locations are displayed in 'Watch-Window' of DSP's IDE, Code Composer Studio.

4.2 Experimental Analysis

First, the velocity of ultrasound in the plain sample is measured by keeping a transmitter and receiver 180° apart. Using that velocity as standard, a set of experiments is performed on echolucent as well as echogenic samples to verify the ultrasound properties analyzed in Chapter 2. Both, echogenic and echolucent, tumors are located at an asymmetric position to confirm that results are valid in a general case. The transmitter is fixed at one position and receiver is moved to various different positions which includes angle of 45°, 90°, 135° and 180° with the transmitter.

The first and second peaks are then used to calculate the location of a tumor and then the findings are compared with actual values to measure the errors between them.

4.2.1 Calculation of Velocity of Ultrasound in Plain Sample

As discussed in Chapter 2, Section 2.2.3, velocity of ultrasound in plain sample is calculated as

$$v = \frac{12.1\text{cm}}{99\mu\text{s}} = 1222.22\text{m/s}$$

This plain sample velocity is further used to calculate the actual location of a tumor.

4.2.2 Echogenic Sample

In this experiment, echogenic mass is embedded in the gelatin as explained in Appendix C to test the different properties and locate the mass from different angles. The actual precise location of the mass is measured from different angles as listed in Table 4.1 to compare the results obtained from the FPGA/DSP co-design system.

Table 4.1 Location of Echogenic Mass from Different Receiver Positions

Angle	Distance from the Mass
0°	3.5cm
25°	3.7cm
45°	3.5cm
90°	4.6cm
135°	5.6cm
180°	5.2cm

Figure 4.1 shows demodulated output of the sample when angle between transmitter and receiver is 180° with its watch-window output in CCS. Table 4.2 lists the actual time required to receive the transmitted signal at different angles for the first as well as second peak noted as $\Delta T1$ and $\Delta T2$. Those times are multiplied by the velocity of

ultrasound calculated in Section 4.2.1 and noted as L1 and L2 respectively. $L_{\text{correction}}$ is the column in which correction is added from the observations in Section 2.2.3. This adds correction to the calculated lengths to determine the actual length. Finally, last column ' ϵ ' calculates the error between actual lengths from Table 4.1 and $L_{\text{corrected}}$.

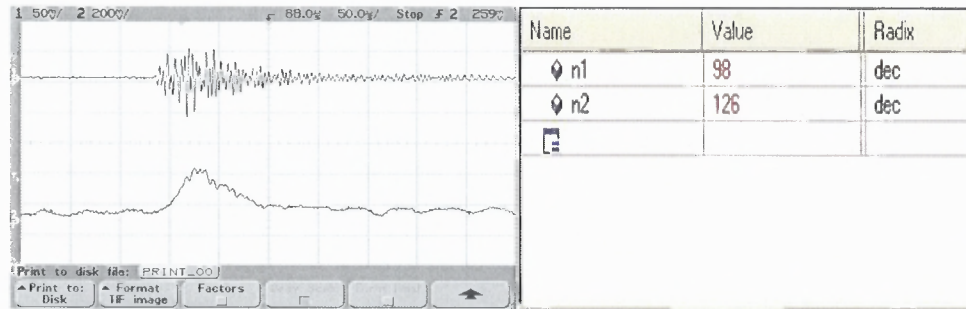


Figure 4.1 Demodulated output at DAC2 with watch-window output in CCS when angle between transmitter and receiver was 180° - echogenic case.

Table 4.2 Calculation of Different Parameters for Echogenic Sample

Angle	$\Delta T1(\mu S)$	$\Delta T2(\mu S)$	L1(cm)	L2(cm)	$L_{\text{corrected}}(\text{cm})$	$\epsilon(\text{cm})$
25°	60	62	7.33	7.58	3.67	0.03
45°	78	117	9.53	14.3	5.86	2.36
90°	88	92	10.76	11.24	4.97	-0.37
135°	99	120	12.09	14.66	5.63	0.03
180°	98	126	11.98	15.4	11.98	0.32

In 45° , 90° and 135° angle cases, $L_{\text{corrected}}$ is calculated by subtracting the already known length of 25° as we are assuming that ultrasound wave is reflected from the surface of mass. As the previous study suggests, 45° case can be ignored as it provides some odd results. In other cases, the error is minimal. The 180° case indicates the distance between the transmitter to the receiver with some delay.

4.2.3 Echolucent Sample

Similar to the echogenic sample, echolucent mass is also embedded in gelatin with a similar procedure in Appendix C. Figure 4.2 shows demodulated output at DAC2 when transmitter and receiver are kept at 180° angle with its watch-window output in CCS. The location of mass measured from different angles at which the receiver is placed is as shown in Table 4.3.

Table 4.3 Distance of Echolucent Mass from Different Positions of Receiver

Angle	Distance of mass from the edge
0°	2.6cm
25°	3cm
45°	3cm
90°	4.6cm
135°	5.3cm
180°	5.4cm

Table 4.4 is a similar table as that of Table 4.2 with the data for the echolucent mass.

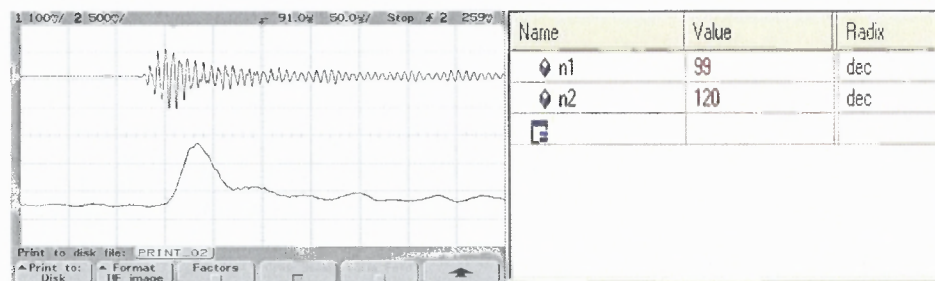


Figure 4.2 Demodulated output at DAC2 with watch-window output in CCS when angle between transmitter and receiver was 180° - echolucent case.

Table 4.4 Calculation of Different Parameters for Echolucent Sample

Angle	$\Delta T1(\mu S)$	$\Delta T2(\mu S)$	L1(cm)	L2(cm)	L1 _{corrected} (cm)
25°	33	179	4.03	21.87	2.02
45°	99	102	12.09	12.47	12.02
90°	87	139	10.63	16.99	7.59
135°	99	121	12.09	14.79	8.64
180°	99	120	12.09	14.67	12.09

In the echolucent case, the time for $\Delta T2$ when the receiver is kept at 25° indicates the full reflection of the ultrasound wave from the opposite end of the sample and hence, when the equivalent length calculated for the time $\Delta T2$ as L2, it is found to be nearly 21.87cm which is nearly twice the length ultrasound wave has to travel when the receiver is kept at 180° from transmitter. No peak between $\Delta T1$ and $\Delta T2$ forces to conclude that either there is no tumor or the tumor, if exists, is benign. In fact, the delay in case of 180° angle, which causes to conclude the longer distance between the transmitter and receiver, indicates that there is some impurity in the path but which is not echogenic but is echolucent. Hence, it is possible to predict the existence of echolucent impurity but hard to locate.

CHAPTER 5

CONCLUSIONS AND FUTURE WORK

5.1 Conclusions

1. The analysis of ultrasound properties of tumors gives deep insight on the detection techniques of tumors since most of the tumors are echogenic in nature.
2. The low transient pulse drive signal causes the generation of short duration of output ultrasound waves at the receiver which can be distinctly identified causing comparatively easy prediction of nature and location of tumors.
3. The FPGA/DSP co-design is implemented in such a way that FPGA runs at higher frequency and carries out low transient pulse drive signal generation at the resolution of 62.5 MHz and detection of the passband signal using QAM demodulator which samples the output signal at 1MHz. The DSP carries out the analysis of demodulated signal to detect the peak location which can further be used to calculate the exact location.
4. The DSP/FPGA platform demonstrates the generation of LTP drive signal as well as detection of the tumor location across the dual platform in a quite efficient way with accuracy. The modular programming approach provides a cost-effective technique of handheld, user-operable, primary and cheaper ultrasonic detection device that detects the locations of tumors with a good precision.

5.2 Areas of Improvement

1. Since the response of ultrasound devices is dependent on various factors, there is a need of standardization. In this work, initially, there are many inconsistencies found in the output due to variable pressure application at the points of contact between transmitter/receiver and sample surface for which a customized assembly is ordered which makes it operator-independent. These parameters are required to be studied before application as a hand-held device.
2. This technique detects echogenic tumors more accurately but there is in depth analysis required for tumors which are echogenic as well as echolucent in nature like Pancreatic Cystadenocarcinoma.
3. A case study is required to eliminate the chances of getting false-positives as well as false-negatives in the output.

5.3 Future Work

1. Work has to be done on small size tumors.
2. An application can be developed, which requires no or little user interaction to detect the exact angle and calculate the location of tumor accordingly.
3. A real-time application of this system has to be tested to determine its accuracy and efficiency.

APPENDIX A

TIMER CONFIGURATION

Source: TMS320C6000 DSP 32-bit Timer Datasheet, SPRU582B, January 2005.

Three registers, Timer Control Register (CTL), Timer Period Register (PRD) and Timer Count Register (CNT) control the timer behavior. CTL determines its operating mode, monitors its status and controls the function of its TOUT pin. PRD contains number of clock input cycles to count which is 32-bit and CNT contains current value of incrementing counter which is also 32-bit.

The sampling period of ADC is set through the 'main' program and the following code is used to configure time period to be counted in PRD register.

```
*(unsigned volatile int *)_TIMER_PRD1_ADDR = 0x01; /* set for 32 bit cnt*/
```

The value is calculated using the formula explained by Equation (3.1) for LTP pulse period. PRD is shown in Figure A.1 which is set with the count and Figure A.2 shows timer control register (CTL).

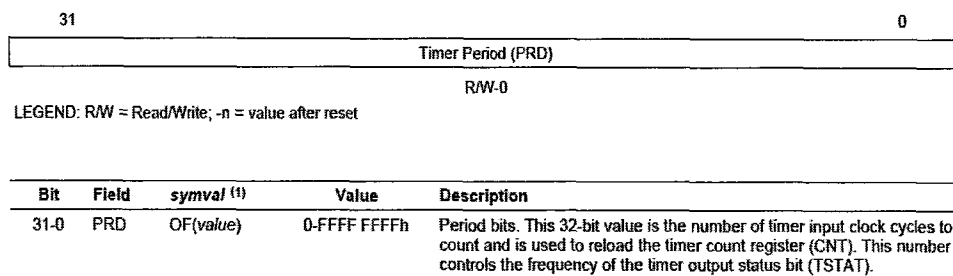
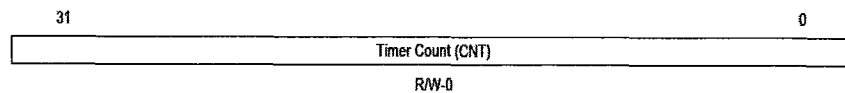


Figure A.1 Timer Period Register Configuration.

Bit	field ⁽¹⁾	symval ⁽¹⁾	Value	Description
31-16	Reserved	-	0	Reserved. The reserved bit location is always read as 0. A value written to this field has no effect.
15	SPND ⁽²⁾		0	Suspend mode bit. Stops timer from counting during an emulation halt. Only affects operation if the clock source is internal, CLKSRC = 1. Reads always return a 0.
			1	Timer continues counting during an emulation halt.
14-12	Reserved	-	0	Reserved. The reserved bit location is always read as 0. A value written to this field has no effect.
11	TSTAT		0	Timer status bit. Value of timer output.
		0	0	
		1	1	
10	INVINP		0	TINP inverter control bit. Only affects operation if CLKSRC = 0.
		NO	0	Noninverted TINP drives timer.
		YES	1	Inverted TINP drives timer.
9	CLKSRC		0	Timer input clock source bit.
		EXTERNAL	0	External clock source drives the TINP pin.
		CUPOVR4	1	Internal clock source. For C62x/C67x DSP: CPU clock/4 For C64x DSP: CPU clock/8
8	CP		0	Clock/pulse mode enable bit.
		PULSE	0	Pulse mode. TSTAT is active one CPU clock after the timer reaches the timer period. PWID determines when it goes inactive.
		CLOCK	1	Clock mode. TSTAT has a 50% duty cycle with each high and low period one countdown period wide.
7	HLD		0	Hold bit. Counter may be read or written regardless of HLD value.
		YES	0	Counter is disabled and held in the current state.
		NO	1	Counter is allowed to count.
6	GO		0	GO bit. Resets and starts the timer counter.
		NO	0	No effect on the timers.
		YES	1	If HLD = 1, the counter register is zeroed and begins counting on the next clock.
5	Reserved	-	0	Reserved. The reserved bit location is always read as 0. A value written to this field has no effect.
4	PWID		0	Pulse width bit. Only used in pulse mode (CP = 0).
		ONE	0	TSTAT goes inactive one timer input clock cycle after the timer counter value equals the timer period value.
		TWO	1	TSTAT goes inactive two timer input clock cycles after the timer counter value equals the timer period value.
3	DATIN		0	Data in bit. Value on TINP pin.
		0	0	Value on TINP pin is logic low.
		1	1	Value on TINP pin is logic high.
2	DATOUT		0	Data output bit.
		0	0	DATOUT is driven on TOUT.
		1	1	TSTAT is driven on TOUT after inversion by INVOUT.
1	INVOUT		0	TOUT inverter control bit (used only if FUNC = 1).
		NO	0	Noninverted TSTAT drives TOUT.
		YES	1	Inverted TSTAT drives TOUT.
0	FUNC		0	Function of TOUT pin.
		GPIO	0	TOUT is a general-purpose output pin.
		TOUT	1	TOUT is a timer output pin.

Figure A.2 Timer Control Register configuration.

Figure A.3 explains the timer count register configuration (CNT) which stores the current count of cycles.



LEGEND: R/W = Read/Write; -n = value after reset

Bit	Field	symbol ⁽¹⁾	Value	Description
31-0	CNT	OF(value)	0-FFFF FFFFh	Main count bits. This 32-bit value is the current count of the main counter. This value is incremented by 1 every input clock cycle.

Figure A.3 Timer Count Register configuration.

Hence, when the period is set like the example below:

`*(unsigned volatile int *)_TIMER_PRD1_ADDR = 0xA0`

The period is $(1010\ 0000)_2 == (192)_{10}$ which sets pulse width of $2.50\ \mu\text{S}$.

APPENDIX B

RESOLUTION

Source: THS1206 12-bit ADC datasheet, SLAS217H, Rev. H, July 2003.

The resolution of output of ADC is calculated as follows.

- Peak to peak voltage at ADC/DAC = $2.7V - (-2.7) = 5.4V$
- ADC output is 12-bit.
- So the 1 bit change in ADC output is equivalent to $5.4V / 2^{12} = 0.0013V$ which is the resolution of output.

If the input voltage to ADC is 1.8V then it is converted to its equivalent digital value as calculated below.

First, the input is divided by the resolution.

$$1.8V / 0.0013 = 1385$$

A DC reference level of 2048 equivalent to $0x008_H$ is added.

$$1385 + 2048 = 3433$$

Then it is converted to digital value.

$$(3433)_{10} = (0x0D69)_H$$

Similarly, an input voltage of -1.8V can be converted as follows:

$$-1.8 / 0.0013 = -1385$$

$$-1385 + 2048 = 663$$

$$663_{10} = 0x0297_H$$

For Digital to Analog conversion, the reverse process takes place.

First a digital value is converted into decimal value.

$$0x0297_H = 663_{10}$$

Then DC reference level equivalent to $0x008_H$ (2048) is subtracted from the decimal value.

$$663 - 2048 = -1385$$

The output is multiplied by resolution of 0.0013 to obtain analog input level.

$$-1385 * .0013 = -1.8V$$

Since the data received is 12-bits and register 16-bits, the incoming sample is divided by 16 and then it is subtracted by 0x008_H.

APPENDIX C

SIMULATED BREAST SAMPLE PREPARATION

Source: www.myscienceproject.org

Ballistic Gelatin is often used by FBI labs to test wound penetration in their ballistic labs. They use special ordnance gelatin powder to make gel to specific standards under carefully controlled conditions and then it is calibrated for accuracy.

The samples used during experimentation are customized for the simulation of human breast. This is a simple homemade procedure that requires Knox plain gelatin, Benefiber fiber supplement and pure cinnamon extract. The procedure can be explained as follows.

- Mix 8 Oz. of Knox Gelatin with one pack of Benefiber in 4 cups of cold water while heating another cup of water on gas. Stir carefully while mixing till it becomes thick.
- Add the boiling water to this mixture and boil it again.
- Put this mixture in refrigerator for a night or at least 3-4 hours.
- Remove the mixture from refrigerator and heat it again on medium heat until it melts. While heating it, keep stirring to avoid bubble formation.
- Add cinnamon drop while stirring to avoid bubbles.
- Pour the heated mixture into 4 bowls. One fully filled and other three half-filled for tumor simulation and keep them in refrigeration for a day.
- Set the tumors provided in 3 half-filled bowls on next day and pour gel on them to keep them fixed at a location. Keep it again in refrigeration for a day.
- On next day, fill the 3 bowls to its full with reheated gel and keep it in refrigeration for three days.
- On the third day, the sample will be ready and can be used for experimentation.

APPENDIX D

EMIF CE SPACE REGISTER CONFIGURATION

Source: TMS320C6000 DSP External Memory Interface (EMIF) Reference Guide, SPRU266E, April 2008.

These registers correspond to CE memory spaces supported by EMIF. There are four CE space control registers and MTYPE field identifies memory type for corresponding CE space. This field should only be set once during the system initialization, except that CE1 is used for the ROM boot mode. The configuration of CECTL0-3 is given in Figure D.1.

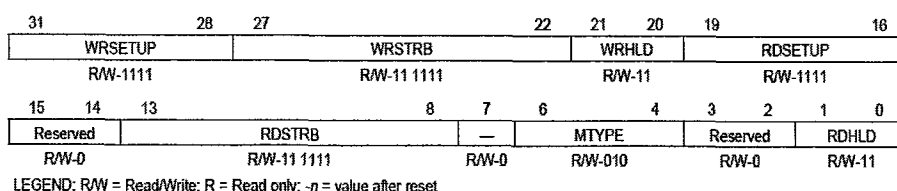


Figure D.1 CECTL0-3 configuration registers.

Table D.1 gives the field description for CECTL registers.

Table D.1 CECTL0-3 Space Control Registers' Field Description

Bit	field	symval	Value	Description
31-28	WRSETUP	OF(value)	0-Fh	Write setup width. Number of clock cycles of setup time for address (EA), chip enable (CE), and byte enables (BE) before write strobe falls. For asynchronous read accesses, this is also the setup time of AOE before ARE falls.
27-22	WRSTRB	OF(value)	0-3Fh	Write strobe width. The width of write strobe (\overline{AWE}) in clock cycles.
21-20	WRHLD	OF(value)	0-3h	Write hold width. Number of clock cycles that address (EA) and byte strobes (\overline{BE}) are held after write strobe rises. For asynchronous read accesses, this is also the hold time of AOE after ARE rising.
19-16	RDSETUP	OF(value)	0-Fh	Read setup width. Number of clock cycles of setup time for address (EA), chip enable (CE), and byte enables (BE) before read strobe falls. For asynchronous read accesses, this is also the setup time of AOE before ARE falls.
15-14	Reserved	-	0	Reserved. The reserved bit location is always read as 0. A value written to this field has no effect.
13-8	RDSTRB	OF(value)	0-3Fh	Read strobe width. The width of read strobe (\overline{ARE}) in clock cycles.
7	Reserved	-	0	Reserved. The reserved bit location is always read as 0. A value written to this field has no effect.
6-4	MTYPE		0-7h	Memory type of the corresponding CE spaces.
		ASYN8	0	8-bit wide asynchronous interface.
		ASYN16	1h	16-bit-wide asynchronous interface.
		ASYN32	2h	32-bit-wide asynchronous interface.
		SDRAM32	3h	32-bit-wide SDRAM.
		SBSRAM32	4h	32-bit-wide SBSRAM.
		-	5h-7h	Reserved.
3-2	Reserved	-	0	Reserved. The reserved bit location is always read as 0. A value written to this field has no effect.
1-0	RDHLD	OF(value)	0-3h	Read hold width. Number of clock cycles that address (EA) and byte strobes (\overline{BE}) are held after read strobe rises. For asynchronous read accesses, this is also the hold time of AOE after ARE rising.

REFERENCES

[1-36]

- [1] (2009), Facts and Statistics About Breast Cancer in the United States: 2009. Available:
http://www.stopbreastcancer.org//index.php?option=com_content&task=view&id=493&Itemid=178
- [2] W. Shao, B. Zhou, Z. Zheng, and G. Wang, "UWB Microwave Imaging for Breast Tumor Detection in Inhomogeneous Tissue," in *IEEE Engineering in Medicine and Biology 27th Annual Conference*, 2005, pp. 1496-1499.
- [3] J. Brockway, K. Subramanian, W. Carruthers, and E. Insko, "Detection and Visualization of Inflammatory Breast Lesions Using Dynamic Contrast Enhanced MRI Volumes," ed. Charlotte, NC: Memory Testing Corporation, 2004.
- [4] P. Huynh, A. Jarolimek, and S. Daye, "The false-negative mammogram," *Radiograph*, vol. 18, pp. 1137-1154, 1998.
- [5] J. G. Elmore, M. B. Barton, V. M. Mocer, S. Polk, P. J. Arena, and S. W. Fletcher, "Ten-year risk of false positive screening mammograms and clinical breast examinations," *New England Journal of Medicine*, vol. 338, pp. 1089-1096, 1998.
- [6] V. P. Jackson, R. E. Hendrick, S. A. Feig, and D. B. Kopans, "Imaging of the radiographically dense breast," *Radiology*, vol. 188, pp. 297-301, August 1993.
- [7] M. Kriege, C. Brekelmans, C. Boetes, P. Besnard, H. Zonderland, I. M. Obdeijn, R. A. Manoliu, T. Kok, and H. Peterse, "Efficacy of MRI and mammography for breast-cancer screening in women with a familial or genetic predisposition," *The New England Journal of Medicine*, vol. 351, pp. 427-437, July 2004.
- [8] B. Cheng and T. Chang, "Enhancing Ultrasonic Imaging with Low Transient Pulse Shaping," *IEEE Transactions on Ultrasonics, Ferroelectrics, and Frequency control*, vol. 54, pp. 627-635, March 2007.
- [9] H. Jiang, C. Li, D. Pearlstone, and L. Fajardo, "Ultrasound-guided microwave imaging of breast cancer - Tissue phantom and pilot clinical experiments," *Medical Physics*, vol. 32, pp. 2528-2535, August 2005.
- [10] D. O. Watermann, M. Foldi, A. Hanjalic-Beck, A. Hasenburg, A. Lughausen, H. Prompler, G. Gitsch, and S. Stickeler, "Three-dimensional ultrasound for the assessment of breast lesions," *Ultrasound Obstet Gynecol*, vol. 25, pp. 592-598, 2005.

- [11] S. Sinha, M. Goodsitt, M. Roubidoux, R. Booi, G. LeCarpentier, C. Lashbrook, K. E. Thomenius, C. L. Chalek, and P. L. Carson, "Automated Ultrasound Scanning on a Dual-Modality Breast Imaging System - Coverage and Motion Issues and Solutions," *Ultrasound in Medicine*, vol. 26, pp. 645-655, 2007.
- [12] (February 2010). Ultrasound and Ultrasonic Testing. Available: <http://www.ndt-ed.org/EducationResources/HighSchool/Sound/ultrasound.htm>
- [13] H. Abu-Rub, J. Guzin'ski, Z. Krzeminski, and H. Toliyat, "Predictive current control of voltage-source inverters," *IEEE Transactions on Industrial Electronics*, vol. 51, pp. 585-593, 2004.
- [14] P. Prieto, M. Sanchez, J. D. Ser, and M. Mendicute, "Design, simulation and hardware validation of a low-cost CLMS echo canceling system," in *2009 Joint IEEE North-East Workshop on Circuits and Systems and TAISA Conference, NEWCAS-TAISA '09*, Toulouse, 2009.
- [15] "Ensemble MTI-203 RapidIO® DSP/FPGA Signal and Video Processing AdvancedMC™ Card," I. Computer Systems, Ed., ed. Chelmsford, MA, 2009.
- [16] D. Daniels and R. Allan, "Multi-channel landmine detection radar signal processing using blind deconvolution," in *Science, Progress and Quality at Radiofrequencies -6th European Radar Conference, EuRAD 2009*, Rome, pp. 97-100.
- [17] M. Brogioli, P. Radosavljevic, and J. Cavallaro, "A general hardware/software co-design methodology for embedded signal processing and multimedia workloads," in *40th Asilomar Conference on Signals, Systems, and Computers, ACSSC '06*, Pacific Grove, CA, 2006, pp. 1486-1490.
- [18] M. Galanis, A. Milidonis, A. Kakarountas, and C. Goutis, "A design flow for speeding-up dsp applications in heterogeneous reconfigurable systems," *Microelectronics Journal*, vol. 37, pp. 554-564, 2006.
- [19] "TMS320C6416T DSK Technical Reference," I. Spectrum Digital, Ed., 508035-0001 Rev. A ed, November 2004.
- [20] "Spartan-IIIE 1.8V FPGA Family: Introduction and Ordering Information," I. Xilinx, Ed., DS077-1 (v1.0) ed, November 2001.
- [21] "THS1206 - 12-bit, 4 Analog Input, 6 MSPS, Simultaneous Sampling Analog-To-Digital Converter," I. Texas Instruments, Ed., SLAS217H ed, July 2003.
- [22] "AD9763/AD9765/AD9767 10-/12-/14-Bit, 125 MSPS Dual TxDAC+ Digital-to-Analog Converters," I. Analog Devices, Ed., Rev. F ed, 2008.
- [23] (2010, March). *VASG: VHDL Analysis and Standardization Group*. Available: <http://www.eda.org/vasg/>

- [24] "R15 α Sensor - General Purpose, 150 kHz Resonant Frequency Acoustic Emission Sensor," Rev. # 123-04 ed: MISTRAS Group Incorporated, March 2010.
- [25] (2010, March). *BDTi DSP Kernel Benchmarks - Speed per Dollar Ratios for Fixed-Point Packaged Processors*. Available:
http://www.bdti.com/products/services_bdti_benchmarks.html
- [26] "TMS320C6414T/ TMS320C6415T/ TMS320C6416T Fixed-Point Digital Signal processors," SPRS226M ed: Texas Instruments, Inc., April 2009.
- [27] "TMS320C6000 DSP 32-Bit Timer Reference Guide," SPRU582B ed: Texas Instruments, Inc., January 2005.
- [28] R. Chassaing, *Digital signal processing and applications with C6713 and C6416 DSK*, Second ed. Hoboken, NJ: A John Wiley & Sons Inc. Publication, January 2005.
- [29] "Code Composer Studio User's Guide," SPRU328B ed: Texas Instruments, Inc., February 2000.
- [30] "TMS320C6000 DSP External Memory Interface (EMIF) Reference Guide," SPRU266E ed: Texas Instruments, Inc., April 2008.
- [31] (2010, March). *Ultrasound*. Available: <http://en.wikipedia.org/wiki/Ultrasound>
- [32] (2010, March). *Tumor*. Available:
<http://www.nlm.nih.gov/medlineplus/ency/article/001310.htm>
- [33] *Medical Ultrasonography*. Available:
http://en.wikipedia.org/wiki/Medical_ultrasonography
- [34] D. Kim and B. G. Lee, "Transform Domain IIR Filtering," *IEEE TRANSACTIONS ON SIGNAL PROCESSING*, vol. 43, pp. 2431-2434, October 1995.
- [35] J.-P. Deschamps, G. J. A. Bioul, and G. Sutter, *Synthesis of Arithmetic Circuits - FPGA, ASIC and Embedded Systems*, 1st ed.: A John Wiley & sons, Inc., Publication.
- [36] A. Mathur, "A FPGA/DSP Design for Real-time Fracture Detection Using Low Transient Pulse," MS, Electrical and Computer Engineering, NJIT, Newark, 2010.